

Weirauch, Michael
Internet Booking Engine mit Datenaggregation aus mehreren unterschiedlich strukturierten Quellen

– DIPLOMARBEIT –

Hochschule Mittweida - University of Applied Science (FH)

Mittweida, 2009

Weirauch, Michael
Internet Booking Engine mit Datenaggregation
aus mehreren unterschiedlich strukturierten
Quellen

eingereicht als – DIPLOMARBEIT –
Hochschule Mittweida - University of Applied Science (FH)

vorgelegte Arbeit wurde verteidigt am 26.08.2009

Erstprüfer	Zweitprüfer
Prof. Dr. Schmalwasser	Dipl.-Kfm. Hoyer

Mittweida, 2009

„Weirauch, Michael:

Internet Booking Engine mit Datenaggregation aus mehreren unterschiedlich strukturierten Quellen. - 2008 - 342 S.

Mittweida, Hochschule Mittweida (FH), Fachbereich Medien, Diplomarbeit“

„Referat

Ziel der Diplomarbeit ist es, eine neue Buchungseingine für das Portal *ferienhaus.de* zu entwickeln. Im Gegensatz zu den am Markt üblichen Portalen soll sich das Angebot nicht nur auf Ferienobjekte eines Anbieters beschränken. Die Engine muss Daten mehrerer Anbieter verarbeiten und sie in einer einheitlichen Form zusammenbringen.

Um Usern einen höheren Anreiz zu geben über dieses Portal zu buchen, wird ihm neben einer großen Auswahl an Objekten ebenfalls ein umfangreiches Repertoire an Merkmalen gegeben, die bei einer Suche berücksichtigt werden. Gleichzeitig sind Privatangebote wie sie bereits auf *ferienhaus.de* existieren eingebunden. Es besteht also ebenfalls ein weiterer Anreiz für Anbieter, die direkt auf *ferienhaus.de* inserieren.

Der erste Teil der Arbeit setzt sich mit dem ist Zustand des *ferienhaus.de* Portals auseinander. Darauf aufbauend wird erläutert, was sich am System ändern soll und wo die Unterschiede des alten zum neuen System liegen. Im Anschluss findet ein Vergleich von mehreren am Markt befindlichen Anbietern und deren Merkmalen statt. Aus den Ergebnissen lässt sich ein Rahmen für das zukünftige Portal und maßgeblich die Booking Engine gewinnen. Es folgt eine Auswertung von Datenformaten und Strukturen der drei beispielhaft für das Projekt herangezogenen Anbieter. Die technischen Parameter für die Realisierung werden gesetzt und benutzte Systeme erklärt. Der Hauptteil handelt die praktische Arbeit ab, dokumentiert diese und zeigt für viele Probleme Lösungswege auf. Abschließend geht es um erzielte Ergebnisse und mögliche Verbesserungen bzw. zukünftige Erweiterungen des Systems“.

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abbildungsverzeichnis	VI
Abkürzungsverzeichnis	VIII
1. Aufgabenstellung	1
2. Ferienhaus.de Zustand 2008	4
2.1. Suchseite Ferienwohnungen	5
2.2. Privatangebote	6
3. Konzept des neuen Systems	8
3.1. Analyse und Auswertung von Konkurrenzportalen	8
3.2. Grundlegende Struktur des eigenen Systems	8
3.3. Anforderungen an die Buchungsmaske	9
3.3.1. Suchmaske	9
3.3.2. Objektübersicht	10
3.3.3. Detailansicht	10
3.3.4. Merkliste	10
3.3.5. Belegungsübersicht	11
3.3.6. Buchung	11
3.4. Datenpool	12
3.4.1. Verteilungsstruktur für Objektdaten	12
3.4.2. Erzeugung eigener Objektcodierung	13
3.4.3. Zusammenfassung von Mehrdeutigkeiten	14
3.4.4. Speicherung von Vorgängen	15
3.5. Aggregation	15
3.6. Kommunikation und Export	16
4. Vergleich von mehreren, stichprobenartig ausgewählten Anbietern	17
4.1. Allgemeines Erscheinungsbild	17
4.2. Suchmaske der Anbieter	18
4.3. Objektübersicht	21
4.4. Detailansicht	24

5. Verwendete Systeme und Software	31
5.1. Entwicklungsumgebung	31
5.2. Angewendete Systeme	32
5.2.1. AJAX	32
5.2.2. PHP	32
5.2.3. SOAP	33
5.2.4. XML	33
5.2.5. SQL	33
5.2.6. HTML	34
6. Datenbank	35
6.1. Einleitung	35
6.2. Tabellentypen	35
6.3. Beschreibung der Tabellen	36
6.3.1. buc_personen	36
6.3.2. menu_land, -region, -ort	36
6.3.3. mon_creditcard	37
6.3.4. mon_wechselkurs	37
6.3.5. obj_buhinweise	38
6.3.6. obj_hauspic, obj_wohnungspic	38
6.3.7. obj_haustext, obj_wohnungstext	39
6.3.8. obj_merkmale	39
6.3.9. obj_preise	40
6.3.10. obj_stammdaten	40
6.3.11. obj_vakanzen	41
6.3.12. obj_zusatzleistungen	41
6.3.13. ref_interhome	42
6.3.14. ref_wolters	42
6.3.15. ref_onferienhaus	43
6.3.16. ref_land, ref_region, ref_ort	43
6.3.17. ref_zusatzleistungen	44
7. Datenformate der Anbieter	45
7.1. Format und Struktur der Interhome Datensätze	45
7.2. Format und Struktur der Wolters Datensätze	46
7.3. Datenformat und Struktur der Privateinträge des ursprünglichen ferien- haus.de Portals	46
8. Datenaggregation	48
8.1. Vorwort	48
8.2. Bezug der Anbieterdaten	48
8.3. Verarbeitung der Daten	50
8.3.1. Bearbeitung von Datensätzen im CSV Format	50
8.3.2. Bearbeitung von Datensätzen im XML Format	54

8.3.3.	Konvertierung von Wolters Datenbeständen	55
8.3.4.	Konvertierung von Stammdaten	57
8.3.5.	Konvertierung Vakanzen von Interhome	59
8.3.6.	Übernahme Privatangebote aus aktuellem in das neue System . .	61
9.	Bedienoberfläche	68
9.1.	Globale Definitionen	68
9.1.1.	Arbeit mit einer Session	68
9.1.2.	Überprüfung von Eingaben	69
9.1.3.	Ansatz zur Objekt orientierten Programmierung	70
9.2.	Suchmaske	71
9.3.	Objektübersicht	74
9.4.	Objekt Detailansicht	78
9.5.	Merkliste	81
9.6.	Belegungsübersicht, Buchungsvorbereitung	82
9.6.1.	Vakanzausgabe Interhome	84
9.6.2.	Vakanzausgabe Wolters	86
9.6.3.	Vakanzausgabe Objekte von Privat	87
9.6.4.	Preisausgabe Interhome	88
9.6.5.	Wolters Preisausgabe	88
9.6.6.	Preisausgabe Privatangebote	89
9.7.	Buchungsmaske	89
9.7.1.	Funktion „Zusatzleistungen Interhome“	92
9.7.2.	Funktion „Zusatzleistungen Wolters“	94
9.8.	Zahlungsangaben	95
9.9.	Eingabeüberprüfung Buchungsmaske	97
9.10.	Darstellung der Angaben	97
9.11.	Übermittlung der Buchungsdaten	99
10.	Erreichte Ziele	104
10.1.	Datenaggregation	104
10.2.	Datenbank	104
10.3.	Buchungssystem	105
10.4.	Schnittstellen	107
10.5.	Export	107
10.6.	Zusätzliche Scripte	107
11.	Zukunftsansichten, Verbesserungen	108
11.1.	Backup Strategien	108
11.2.	Stammdatenroutine	108
11.3.	Objektcodierung	109
11.4.	Sortieroptionen in der Übersicht	109
11.5.	Objektanzahl in der Suchmaske	109
11.6.	Verändertes Verfahren der Datenausgabe	110

11.7. Buchungsarten	110
11.8. Sicherheit	110
11.9. Reseller Option	110
12. Resümee	111
12.1. Die Aufgabe	111
12.2. Ergebnisse	111
Literaturverzeichnis	114
Selbstständigkeitserklärung	332

Abbildungsverzeichnis

1.1. Ferienhaus.de, alte Buchungsvariante	2
1.2. Ferienhaus.de, neue Buchungsvariante	3
2.1. Ferienhaus.de Startseite	4
2.2. Ferienhaus.de Suchseite	5
2.3. Übersicht Privatangebote	6
2.4. Detailansicht Privatangebote	7
4.1. Standardsuche, e-domizil.de	18
4.2. Volltextsuche, e-domizil.de	19
4.3. Erweiterte Suche, e-domizil.de	19
4.4. Suchmaske, tourist-online.de	20
4.5. Regionssuche, tourist-online.de	20
4.6. Suchmaske erweiterte Auswahl, interchalet.com	21
4.7. Objektübersicht, e-domizil.de	22
4.8. Objektübersicht, Tourist-online.de	22
4.9. Übersicht, interchalet.com	23
4.10. Übersicht, ferienhaus.com	24
4.11. Detailansicht Objekt, e-domizil.de	25
4.12. Detailansicht Kalender, e-domizil.de	26
4.13. Detailansicht, tourist-online.de	27
4.14. Auszug aus Detailansicht, interchalet.com	28
4.15. Detailansicht Auszug 1, ferienhaus.com	29
4.16. Detailansicht Auszug 2, ferienhaus.com	30
8.1. Ablauf der Datenaggregation	49
8.2. Auszug aus der Tabelle ref_land	58
8.3. Ablauf Stammdatenkonvertierung	67
9.1. Session, Datenverteilung	69
9.2. Suchmaske, Profisuche	72
9.3. Programmablauf Suchmaske	73
9.4. Objektübersicht	74
9.5. Detailansicht eines Objekte	78
9.6. Ablauf Detailansicht	80
9.7. Merkliste	81
9.8. Belegungsübersicht	83

9.9. Header und erster Teil des Formulars	89
9.10. Auswahl von Leistungen und Zahlungsart	90
9.11. Ablauf Buchung Teil 1	91
9.12. Eingabe von Zahlungsinformationen, hier Kreditkartenzahlung	96
9.13. Ausgabe der erfassten Daten	98
9.14. Ablauf Datenausgabe	99
9.15. Ablauf Datenversand	103

Abkürzungsverzeichnis

CSV	Comma Separated Value
HTML	Hypertext Markup Language
IBE	Internet Booking Engine
MiB	Mebibyte
OOP	Objektorientierte Programmierung
ODBC	Open Database Connectivity
PHP	HyperText Preprocessor
SQL	Structured Query Language
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
WWW	World Wide Web

1. Aufgabenstellung

In Folge der Modernisierung einiger Portale des Unternehmens Online Now! steht u.a. auch ferienhaus.de auf dem Prüfstand. Der momentane Internetauftritt ist laut Aussagen des Inhabers¹ nicht mehr zeitgemäß. Auf der Plattform können Anbieter von Ferienobjekten inserieren. Diese sind dann direkt im System gespeichert. Um einen größeren Datenbestand anzubieten laufen Verträge mit zwei großen Anbietern deren Objekte ebenfalls über Ferienhaus.de buchbar sind. Eingebunden sind diese Angebote jedoch nur über ein Iframe². Die Buchungsmaske liegt auf den Servern der Anbieter.

Es ist ein Relaunch der Seite geplant. Wichtiger Bestandteil der Planungen ist der Aufbau eines eigenen Datenpools und Buchungssystems. Der Datenpool enthält dabei nicht nur die privaten Angebote wie sie momentan auf ferienhaus.de zu finden sind. Datenbestände der Anbieter, welche zur Zeit über die Iframe Lösung angebunden sind, sollen ebenfalls im Datenpool bereit stehen. Dies erspart das Einbinden verschiedener Anbieter auf separaten Seiten. Alle Daten lassen sich ab diesem Zeitpunkt über ein und dieselbe Buchungsmaske abrufen.

Es ist gefordert, dass Aktualisierungen der Datenbestände im Datenpool automatisiert und zeitgesteuert ablaufen. Der Bereich umfasst den Download, Konvertierung, Referenzierung und Update der Datensätze. Überprüfungsrouitinen müssen dabei jeden Schritt überwachen, ein Logbuch anlegen und im Fehlerfall den aktuellen Vorgang abbrechen und eine Nachricht an den Administrator versenden. Buchungsvorgänge sollen ohne manuellen Eingriff des Portalanbieters möglich sein. Bucht der User ein Objekt, werden seine Daten direkt an den betreffenden Anbieter weitergeleitet.

Im Mittelpunkt der Arbeit steht deshalb die Realisierung der Buchungsmaske und des Datenpools sowie dessen automatisierte Pflege. Das Feld erstreckt sich über:

- Erfassung der Daten
- Konvertierung auf einheitliche Strukturen
- Aufbau der Datenbank, dient als Datenpool
- Anwendung der Daten in der Buchungsmaske
- Versand von Buchungsdaten an die Anbieter

Gestalterische Elemente wie beispielsweise der Grafische Aufbau der Buchungsmaske spielen dabei nur eine untergeordnete Rolle. Die endgültige Gestaltung wird von der

¹ Dies war der zweite wichtige Punkt neben dem Aufbau eines eigenen Buchungssystems. Er wurde zu Beginn der Arbeiten erwähnt und während einer Besprechung im April 2009 nochmals bestätigt.

² Skop 2007, Javascript/DOM - Objektreferenz - HTML-Elementobjekte - Iframe

Firmen-internen Grafik Abteilung vorgenommen. Das momentane Layout, wie es auch in Illustrationen der Diplomarbeit zu finden ist, dient nur als Anhaltspunkt.

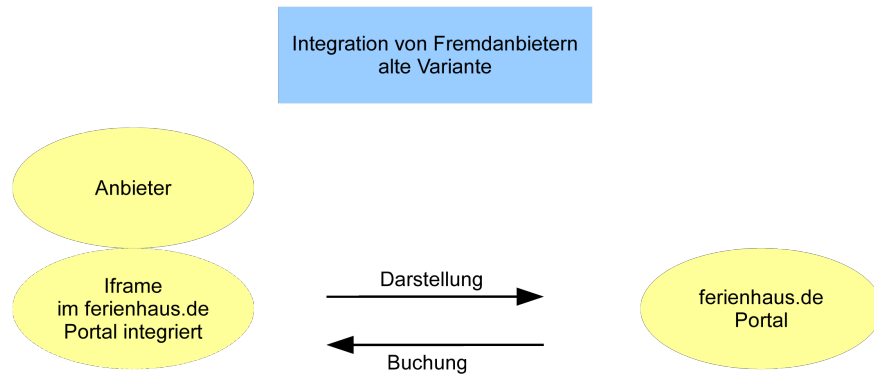


Abbildung 1.1.: Ferienhaus.de, alte Buchungsvariante

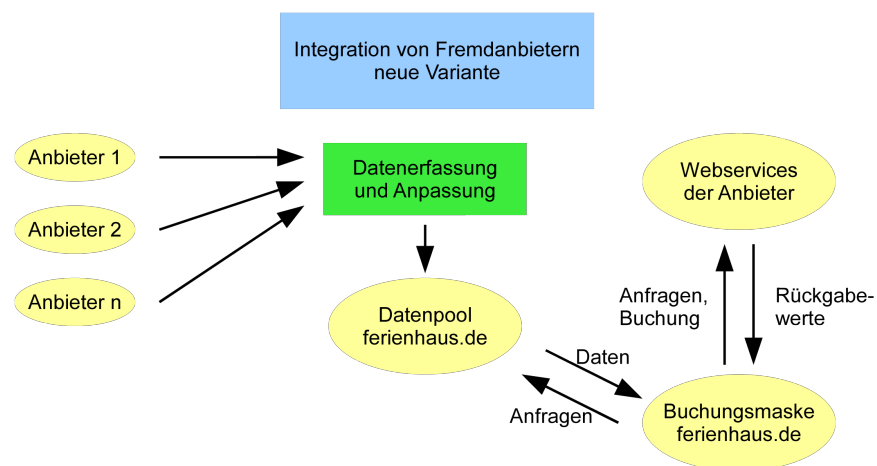


Abbildung 1.2.: Ferienhaus.de, neue Buchungsvariante

2. Ferienhaus.de Zustand 2008

Ferienhaus.de bietet Häuser von Privatinserenten und zwei externen Anbietern die über Iframes³ eingebunden sind. Die Angebote sind nicht gebündelt und stehen auf separaten Seiten auf dem Portal zur Verfügung. Die Ferienhäuser des Anbieters *Interhome*⁴ lassen sich gleich durch drei Links im Navigationsmenü auf der linken Seite aufrufen. Eine Verlinkung der privaten Angebote findet sich dagegen nur im Kopf der Seite. Durch die Gestaltung ist er kaum als Navigator erkennbar. Zumindest fiel dem Autor dieser Diplomarbeit in der ersten Zeit nicht auf, dass die Kopfzeile keine Aufzählung, sondern eine Navigation ist. Die Masse an Links im Content Bereich lenkt vom Header ab. Folglich ist es wahrscheinlicher, dass sich ein Besucher durch den Ferienhaus Bestand von Interhome klickt und die Privatangebote übersieht.

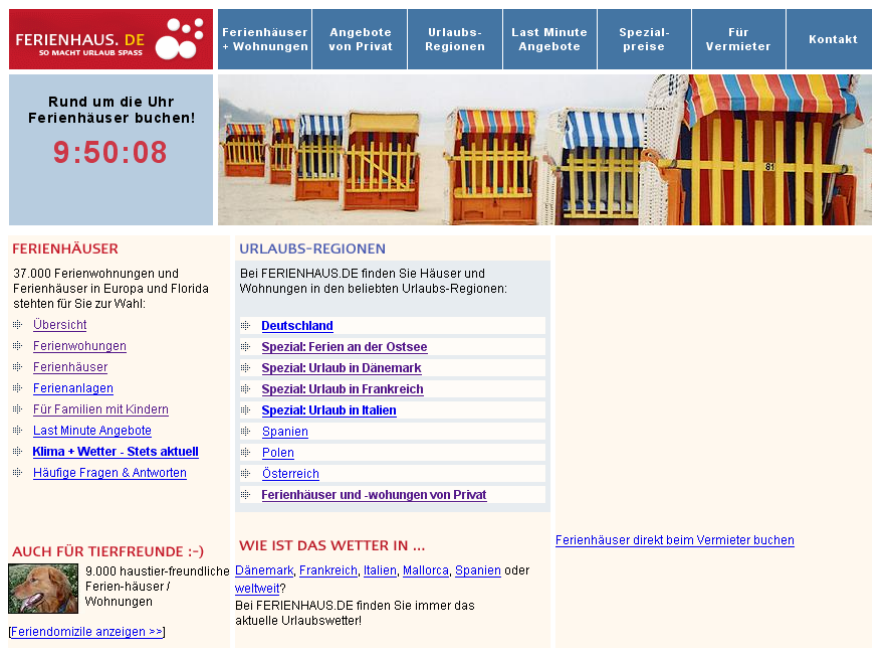


Abbildung 2.1.: Ferienhaus.de Startseite⁵

³ Lubkowitz 2005, 250ff

⁴ <http://www.interhome.de>

⁵ <http://www.ferienhaus.de>

2.1. Suchseite Ferienwohnungen

Das Navigationsmenü der Suchseite enthält nun die Möglichkeit sowohl private Angebote⁶ als auch die Angebote des zweiten externen Anbieters *Novasol*⁷ aufzurufen. Die Seite enthält allerdings schon die Suchmaske des Anbieters Interhome. Die Gefahr ist groß, dass sich ein Besucher auch hier sofort auf die Suchmaske konzentriert und das Navigationsmenü nicht wahr nimmt.

FERIENHAUS.DE
SO MACHT URLAUB SPASS

Rund um die Uhr Ferienhäuser buchen!
9:50:53

FERIENHÄUSER

- [Ferienwohnungen](#)
- [Ferienhäuser](#)
- [Ferienhaus von privat](#)
- [Ferienhäuser im Skigebiet](#)
- [Speziell an der Ostsee](#)
- [Speziell in Dänemark](#)
- [Speziell in Frankreich](#)
- [Für Vermieter](#)
- [FAQ](#)

FERIENHAUS.DE
Buchung & Beratung
030/ 615 082 185
(gebührenfreie Rufnummer)
werktags von 9 - 14 Uhr
Nutzen Sie auch unser
Anfrageformular!

[Ferienhäuser direkt beim Vermieter buchen](#)

Hier haben Sie die freie Auswahl zwischen 37.000 Ferienwohnungen, Ferienhäusern und Apartments in Deutschland, Österreich, Italien, der Schweiz und weiteren europäischen Ländern sowie Florida.

Für Buchungen stehen Ihnen unsere Ferienhaus-Berater auch unter der **kostenfreien Rufnummer +49 30 615 082 185** (werktags von 09.00 Uhr - 14.00 Uhr) zur Verfügung.

Auch für [Gäste mit Haustieren](#) haben wir über **9.000 Ferienobjekte** für Ihren nächsten Urlaub:

Merkliste: 0 Objekte - Jetzt Merkliste zusammenstellen

Startseite > Suchen & Buchen

Standardsuche **Profisuche** **Kartensuche**

Wird geladen. Bitte warten.

Region

Ort

Qualitäts (Minimum):

Aufenthaltsdauer

Anzahl Personen

suchen

Ausstattung

Anzahl Zimmer: von bis

Anzahl getr. Schlafzimmer: von bis

Haus typ:

Schwimmbad ☐

Haustier erlaubt ☐

Kinderbett ☐

Geschirrspüler ☐

Klimaanlage ☐

Waschmaschine ☐

TV ☐

Sauna ☐

Lage

Nähe zum Meer/See:

Nähe zum Skilift:

Entfernung Ortszentrum:

Aktivitäten

Nightlife ☐

Tennisplatz ☐

Surfen ☐

Segeln ☐

Mountain-Biking ☐

Nähe zum Golfplatz:

Reiten ☐

Preis

Maximaler Preis pro Woche:

Angebotslinie:

Abgeschlossenheit ☐

Aussergewöhnliches Objekt ☐

[Ferienhäuser Last Minute buchen](#)

suchen

Abbildung 2.2.: Ferienhaus.de Suchseite⁸

⁶ <http://www.ferienhaus.de/ferienhaeuser/index.php>

⁷ <http://www.novasol.de/>

⁸ <http://www.ferienhaus.de/ferienhaus/ferienwohnungen.php>

2.2. Privatangebote

Findet der Besucher die Privatangebote, erwartet ihn eine Übersicht, welche alle verfügbaren Objekte mit Kurzbeschreibung anzeigt. Leider gibt es keine Begrenzung auf eine bestimmte Anzahl von Objekten, die pro Seite ausgegeben werden. Die Seite ist dadurch unheimlich lang. Sollte die Anzahl an Objekten weiter steigen, nimmt die Unübersichtlichkeit zu und das Scrollen auf der Seite erschwert sich. Gut ist dagegen die Möglichkeit, nur Ferienobjekte eines bestimmten Landes anzuzeigen zu lassen. Innerhalb des Landes ist die Eingrenzung auf eine Region möglich.








FERIENHAUS.DE SO NACHT URLAUB SPASS	Ferienhäuser + Wohnungen	Angebote von Privat	Urlaubs- Regionen	Last Minute Angebote	Spezial- preise	Kontakt
Rund um die Uhr Ferienhäuser buchen! 9:58:20						
FERIENHÄUSER VON PRIVAT zurück zur Länderübersicht Deutschland Allgäu - Ferienhaus in Woringen Bayern - Ferienhaus "Wirtshäusle" - Franks Ferienvilla - Landhaus Wild - Schauberger Hütte Berlin - Ferienhaus in Stolpe-Süd - Ferienhaus Kemna - Ferienwohnung in Marzahn Brandenburg - Ferienhaus "Haus Ella" - Ferienhaus in Bad Saarow - Ferienhaus in Zehdenick - Schlossappartements in Zehdenick - Schlossappartement Typ 2, in Zehdenick - Schlossappartement Typ 3, in Zehdenick Hessen - Ferienhaus Fachwerk und Mehr Mecklenburg - Vorpommern - Ferienhof "Strandkiele" - Finnhütten - Ferienhof "Storchennest" - Ferienhaus in Greifswald - Ferienhaus in Wokuhl Niedersachsen - Ferienhaus in Gräpel - Ferienhaus in Lindwedel - Haus Wattwurm - Karl's Ferienhaus im Oberharz Nordsee/Ostsee - Dat Iuttje Huuske /Greetsiel - Neuwesteel - Ferienhof Puddeminer-Wiek - Ferienhaus "Windlandhütte" - Ferienhaus am Jagdweg - Ferienhaus Gasteiger, Varel-südlicher Jadebusen - Ferienhaus in Glücksburg, Ostsee - Ferienhaus in Koserow		Ferienhaus in Lindwedel, Niedersachsen, Deutschland  stilvolles, romantisches Niedersachsenhaus – Wohnteil eines ehemaligen Bauernhauses mit großem Garten und Terrasse Weitere Informationen + Bilder				
		Ferienhaus in Woringen, Allgäu ****, Allgäu, Deutschland  Optimal für 2 bis 5 Feriengäste, erwarten Sie ein großes Wohn- Esszimmer, WC, eine komplett ausgestattete Küche mit Geschirrspüler, Mikrowelle, Ceran-Kochfeld, Backofen, Küchenmaschine, Besteck, Geschirr, Kühlschrank mit 3Sterne-Gefrierfach usw. Waschmaschine/Trockner, Bügelbrett & -eisen. Weitere Informationen + Bilder Buchen				
		Ferienhaus in Dankerode, Sachsen-Anhalt, Deutschland  Das Blockhaus aus nordischer Fichte (Stammdurchmesser 24 cm) bietet Ihnen im Untergeschoss einen barrierefreien Wohnraum von 70m². Weitere Informationen + Bilder				
		Ferienhof Puddeminer Wiek ****, Nordsee/Ostsee, Deutschland  Unser Ferienhaus Malve 3 in Puddeminer auf Rügen/Ostsee befindet sich in einem gemütlichen Ferienhof, das insgesamt 14 Häuser umfasst und in einem ca. 20.000 grossen Biosphärenreservat liegt. Der Ort strahlt Ruhe und Gemütlichkeit aus. Ideal um Wanderungen und Radtouren zu machen. An allen Küsten Rügens gibt es viele schöne Strände zum Baden und Verweilen. Weitere Informationen + Bilder				
		Karl's Ferienhaus ****, Niedersachsen, Deutschland  Karl's Ferienhaus bietet auf 240qm Platz für bis zu 16 Personen. Die Räumlichkeiten bestehen aus einem Esszimmer, einem Wohnraum, einer offenen Küche, sowie 8 Schlafräumen und 3 Bädern. Weitere Informationen + Bilder Buchen				
		Dat Iuttje Huuske /Greetsiel Neuwesteel, Nordsee/Ostsee, Deutschland  Wir geben Ihnen einen kleinen Einblick in das, was Ihre Erwartungen bestimmt voll erfüllt! Unser Friesenhaus - ein freistehendes Haus mit 80 qm Wohnfläche für 4 Personen geeignet! Weitere Informationen + Bilder				

Abbildung 2.3.: Übersicht Privatangebote⁹

⁹<http://www.ferienhaus.de/ferienhaeuser/index.php>

Zu bemängeln ist an dieser Stelle die fehlende Suchfunktion. Aufgrund der überschaubaren Anzahl an Ferienobjekten stellt dies noch kein großes Problem dar. Mit steigender Anzahl an Objekten erweist sich eine Suchfunktion jedoch als Vorteil. Die Suche könnte zudem eine Möglichkeit darstellen, den Zugriff auf private Angebote auf allen Seiten des Portals zu realisieren.

Jedes Objekt besitzt eine Detailansicht. Sie fällt großzügig aus und enthält eine Vielzahl von Informationen. Die Auflistung von Merkmalen ist umfassend. Eine Beschreibung und die Bildergalerie runden das Angebot ab. Über die Navigation sind weitere Informationen abrufbar. Alles in allem gibt es ein umfassendes Informationsangebot.

Die Buchung der Objekte ist allerdings nur über persönlichen Kontakt mit den Vermietern möglich. Ein Automatisiertes Buchungssystem gibt es nicht.








FERIENHAUS.DE <small>SO MACHT URLAUB SPASS</small>	Ferienhäuser + Wohnungen	Angebote von Privat	Urlaubs- Regionen	Last Minute Angebote	Spezial- preise	Kontakt		
Rund um die Uhr Ferienhäuser buchen! 9:54:25								
FERIENHÄUSER VON PRIVAT		Ferienhaus in Greetsiel / Neuwesteel						
zurück zur Landesübersicht Objektvorstellung <ul style="list-style-type: none"> - Übersicht - Freizeit / Sport - Umgebung - Belegungsübersicht - Bildergalerie - E-Mail 		<div data-bbox="608 927 858 1187">  </div> <div data-bbox="863 927 1201 1187"> Kontakt zum Vermieter Herr Rolf Thomann Dat luttje Huuske /Greetsiel /Neuwesteel Schulweg 91 DE-26506 Neuwesteel Deutschland Telefon: 04934/990329 Telefax: 04934/990329 E-Mail an den Vermieter </div>						
Galerievorschau      Für Vermieter		<div data-bbox="608 1193 901 1299"> Auf einen Blick Wir geben Ihnen einen kleinen Einblick in das, was Ihre Erwartungen bestimmt voll erfüllt! Unser Friesenhaus - ein freistehendes Haus mit 80 qm Wohnfläche für 4 Personen geeignet! </div> <div data-bbox="906 1193 1201 1321"> Vermietungstarife Vom 1.Juni-31.Oktober 2006 65,-€ pro Tag 1.November-31.Dezember 2006 49,-€ pro Tag Hund oder Katze 2,-€ pro Tag Bettwäsche 10,-€ pro Person </div> <div data-bbox="608 1328 1201 1704"> <table border="1"> <tr> <td data-bbox="608 1328 901 1704"> Daten Objekttyp: Haus Maximale Belegung 4 Erwachsene 4 Betten insgesamt Fläche 80 qm Wohnfläche 800 qm Gesamtfläche Räumlichkeiten 2 Schlafzimmer (für Erwachsene und /oder Kinder geeignet) 1 Wohnzimmer 1 Esszimmer 1 Küche 1 Badezimmer 1 Gäste-Toilette 1 Abstellraum Kategorisierung Haustiere erlaubt </td> <td data-bbox="906 1328 1201 1704"> Ausstattung Zentralheizung Wohnraum Bodenbelag: Fliesen SAT-TV Stereoanlage Gesellschaftsspiele 1 Kinderstuhl Küche Geschirrspüler Kühlschrank Gefrierschrank Kaffeemaschine Stromherd Backofen Badezimmer Badewanne extra groß Dusche </td> </tr> </table> </div>					Daten Objekttyp: Haus Maximale Belegung 4 Erwachsene 4 Betten insgesamt Fläche 80 qm Wohnfläche 800 qm Gesamtfläche Räumlichkeiten 2 Schlafzimmer (für Erwachsene und /oder Kinder geeignet) 1 Wohnzimmer 1 Esszimmer 1 Küche 1 Badezimmer 1 Gäste-Toilette 1 Abstellraum Kategorisierung Haustiere erlaubt	Ausstattung Zentralheizung Wohnraum Bodenbelag: Fliesen SAT-TV Stereoanlage Gesellschaftsspiele 1 Kinderstuhl Küche Geschirrspüler Kühlschrank Gefrierschrank Kaffeemaschine Stromherd Backofen Badezimmer Badewanne extra groß Dusche
Daten Objekttyp: Haus Maximale Belegung 4 Erwachsene 4 Betten insgesamt Fläche 80 qm Wohnfläche 800 qm Gesamtfläche Räumlichkeiten 2 Schlafzimmer (für Erwachsene und /oder Kinder geeignet) 1 Wohnzimmer 1 Esszimmer 1 Küche 1 Badezimmer 1 Gäste-Toilette 1 Abstellraum Kategorisierung Haustiere erlaubt	Ausstattung Zentralheizung Wohnraum Bodenbelag: Fliesen SAT-TV Stereoanlage Gesellschaftsspiele 1 Kinderstuhl Küche Geschirrspüler Kühlschrank Gefrierschrank Kaffeemaschine Stromherd Backofen Badezimmer Badewanne extra groß Dusche							

Abbildung 2.4.: Detailansicht Privatangebote¹⁰

¹⁰<http://www.ferienhaus.de/ferienhaeuser/index.php> - Anwahl des gewünschten Objektes

3. Konzept des neuen Systems

3.1. Analyse und Auswertung von Konkurrenzportalen

Bevor der Relaunch stattfindet, muss der aktuelle Markt analysiert werden. Ohne diesen Schritt ist zwar auch ein Relaunch möglich, jedoch besteht die Gefahr, dass am Markt vorbei entwickelt wird. Dabei werfen sich folgende Fragen auf.

- Wie ist die Engine anderer Marktteilnehmer strukturiert?
- Welche Möglichkeiten der Suche gibt es?
- Wie präsentieren sich die Ergebnisse?
- Welche Angaben sind in den Ergebnissen zu finden?
- Gibt es nützliche Zusatzfunktionen?
- Welche Möglichkeiten der Buchung gibt es?

Hierbei können nur offensichtliche Aspekte ausgewertet werden. Im Hintergrund stattfindende Prozesse wie Anbindung an die Datenbank, Ablaufsteuerung und Generierung sind nicht einsehbar. Dies ist auch nicht nötig, da es während der Analyse nur darum geht, einen Überblick über Möglichkeiten und Gestaltung der Systeme anderer Anbieter zu bekommen. Gleichzeitig sind u.U. auf den Seiten des ein oder anderen Anbieters Fehler erkennbar, welche sich im neu zu entwickelnden System von vornherein vermeiden lassen.

3.2. Grundlegende Struktur des eigenen Systems

Als grundlegende Vorgabe für das neue ferienhaus.de Portal dient ein Konzept, welches vorsieht die sog. IBE in drei Teile zu gliedern. Dabei handelt es sich um:

- Datenpool: Zentraler Punkt im System. Enthält die Daten aller Anbieter welche mit ferienhaus.de zusammenarbeiten.
- Datenabholung: Zeitgesteuerter Prozess der Anbieterdaten in den Datenpool einträgt und für stets aktuelle Daten im System sorgt.
- Buchungsmaske: Eine Oberfläche über die Objekte gesucht und gebucht werden können.

3.3. Anforderungen an die Buchungsmaske

Aus internen Besprechungen, den Ergebnissen der Analyse und dem Aspekt das mehrere Anbieter gebündelt werden, geht folgender Entwurf für die Buchungsmaske hervor.

- Traditioneller Aufbau, d.h. der Suchmaske folgt eine Übersicht. Auswahl eines Objektes in der Übersicht führt zur Detailansicht. Detailansicht verlinkt auf Buchungsmöglichkeit.
- System muss die verschiedenen Schnittstellen der Anbieter beherrschen
- Die Verarbeitung von Daten, speziell Aufruf von Vakanzen und Versenden der Daten, muss auf den jeweiligen Anbieter zugeschnitten sein. Der User darf dabei nicht bemerken, dass unterschiedliche Anbieter angesprochen werden.
- Jede Seite enthält eine Navigation durch die auf bestimmte Teile der Buchungsmaske zurückgegriffen werden kann.
- Neben clientseitiger - auch serverseitige Kontrolle der Eingaben erforderlich. Dies betrifft alle Eingaben, die ein User im gesamten Buchungssystem tätigen kann!
- Eingaben, die der User tätigt, müssen für den Zeitraum seiner Verweildauer in einem Zwischenspeicher gehalten werden.
- Logische Strukturierung der zwischengespeicherten Daten
- Die Buchungsmaske muss auch auf Client-Systemen mit eingeschränkter Funktionalität lauffähig sein

3.3.1. Suchmaske

- Die Suchmaske ist in zwei Varianten auszuführen, einer Standard und einer Profisuche
- In beiden Suchvarianten muss die Suche über Auswahl von Land, Region, Ort bzw. Direkteingabe möglich sein. Eine direkte Suche nach bestimmten Objekten ist gefordert. Angaben wie Anreisedatum, Aufenthaltsdauer und die Möglichkeit nur Privatangebote zu durchsuchen sind ebenfalls zu realisieren.
- In der Profisuche sollen zusätzliche Ausstattungsmerkmale, Extras, Personen-, Zimmer- und Schlafräumenzahl auswählbar sein
- Eine Suche ohne getroffene Auswahl darf nicht möglich sein
- Eine Suche ohne Ergebnisse darf nicht auf die Objektübersicht verweisen

3.3.2. Objektübersicht

- In der Objektübersicht ist eine Kurzbeschreibung, Merkmale des Objektes, Preisangaben sowie Vorschaubild und Objektcode gefordert.
- Die Objektübersicht listet Standardmäßig 10 Objekte. Die Anzahl an Objekten pro Seite muss wählbar sein!
- Die Anzahl anwählbarer Seiten ist auf 10 zu begrenzen. Sollten mehr als 10 Seiten vorhanden sein, ist der Seitenzeiger dynamisch anzupassen
- Optional kann eine Sortierung nach Preis oder Objekt erfolgen.

3.3.3. Detailansicht

- Geografische Angaben zum Objekt
- Dem User die Möglichkeit geben das Objekt zu buchen oder in eine Merkliste aufzunehmen
- Besonderer Wert muss auf die Bildergalerie gelegt werden, sie ist das Aushängeschild des Ferienobjektes!
- Alle Merkmale, sowie Angaben über Räumlichkeiten und evtl. vorhandene Bewertungen, müssen in der Ansicht zu sehen sein.
- Galerie, Merkmale und geografische Lage sind als erstes aufzuführen, Beschreibungstexte folgen im 2. Teil

3.3.4. Merkliste

- Speicherung der Verweise zu den Einträgen lokal auf dem Rechner des Users
- Ausgabe soll dem Bild der Objektübersicht entsprechen
- Funktionen, um Objekte wieder aus der Liste löschen zu können, müssen integriert sein
- Vorhaltezeit der gemerkten Objekte auf 2 Wochen begrenzen
- Eintragung von Objekten muss im Hintergrund geschehen

3.3.5. Belegungsübersicht

- Kommunikation mit mehreren Anbietern und deren Schnittstellen
- Erkennung von welchem Anbieter das Objekt stammt, Ausgaben müssen sich an deren Spezifikationen orientieren
- Belegungsübersicht muss Einträge ab dem aktuellen Tag bis zu einem Jahr im Voraus anzeigen können
- Dabei ist die Belegung für den Aktuell gewählten und den Folgemonat anzuzeigen
- Alle möglichen Anreisetage müssen wählbar sein
- Bei Auswahl eines Anreisetages sind Preise, ausgehend von den Konditionen des Anbieters, aufzuführen.
- Preise sind anwählbar
- Bei Auswahl eines Preises soll das aktuelle Anreisedatum mit dem Datum überschrieben werden, welches aus dem Kalender gewählt wurde und für das die Konditionen und der Preis gilt.
- Sind Buchungsdatum und Anreisedatum zu nahe¹¹ beieinander, darf die Buchung nicht möglich sein.
- Dem User müssen aktuell gewähltes Anreisedatum und Aufenthaltsdauer zu Kontrollzwecken angezeigt werden.
- Die Art der Belegung muss gekennzeichnet sein. Bevorzugte Variante ist dabei die farbliche Unterteilung - inklusive Legende.

3.3.6. Buchung

1. Erfassung Personendaten mit Wahl der Zusatzleistungen und Zahlungsart sowie Bestätigung von Hinweisen und AGBs
 - Erfassung der Daten, Name, Vorname, Adresse u.s.w..
 - In der Auswahl des Herkunftslandes die gebräuchlichsten Länder an erster Stelle
 - Die max. auswählbare Anzahl an Reisenden darf nicht die max. Belegung des Objektes überschreiten
 - Ausgaben, wie Zusatzleistungen und Zahlungsart, sind Anbietergebunden und müssen entsprechend nach den Kriterien des Anbieters eingefügt sein
 - Zu jedem Anbieter ist ein Verweis auf dessen AGBs Pflicht!

¹¹ Die mindestens einzuhaltenen Zeitspannen sind in den Unterlagen der Anbieter definiert.

- Sowohl AGB als auch Buchungshinweise müssen erst bestätigt werden, bevor der Buchungsvorgang weiter gehen kann.
2. Erfassung der Zahlungsdaten
 - Hat sich nach der Auswahl, welche auf der vorherigen Seite getroffen wurde, zu richten
 - Bei Kreditkarten sind nur diese anzugeben, welche vom Anbieter unterstützt werden
 - Das Verfallsdatum der Kreditkarte sollte über ein Menü wählbar sein
 3. Übersicht aller Eingaben zur userseitigen Kontrolle
 - Ausgabe aller vom User getätigten Eingaben
 - Angabe von Objektpreis, Zusatzleistungen, Kautions, Objektkennung, Anreisedatum und Aufenthaltsdauer
 - Berechnung von Zusatzleistungen nach Berechnungsart und Zusammenfassen der Ergebnisse
 - Berechnung der Währung in Euro und Angabe der eigentlichen Währung neben der berechneten, wenn nicht schon als Euro vom Anbieter geliefert
 - Möglichkeit Angaben abzusenden oder eine Korrektur vorzunehmen
 4. Datenversand an den jeweiligen Anbieter
 - Überprüfung Belegungsstatus des Objektes durch direkte Abfrage der Anbieterschnittstelle
 - Aufbereitung der Daten nach den Spezifikationen des entsprechenden Anbieters
 - Versenden über dessen Schnittstelle
 - Empfang und Auswertung der Rückgabewerte vom Anbieter
 - Ausgang der Buchung mit Zeitstempel und Angaben zur buchenden Person in der Datenbank festhalten
 - Je nach Fall: Dem User eine Erfolgs- bzw. Fehlermeldung präsentieren

3.4. Datenpool

3.4.1. Verteilungsstruktur für Objektdaten

Dem Aufbau der Datenbank geht eine Analyse der Anbieterdaten voraus. Sie sind nach bestimmten Kriterien strukturiert. Anhand der Strukturen lässt sich ein Bild der Datenbank aufstellen. Alle Anbieter haben gemein, dass sich Datensätze über die Stammdaten referenzieren lassen. Stammdaten bestehen aus der Referenznummer des Objektes und dem zugehörigen Land, Region und Ort bzw. einer Codierung dieser Werte, welche sich

über die zugehörigen Dokumente decodieren lassen. Daneben werden in einer weiteren Datei oder zusammen mit den Stammdaten alle Merkmale des Objektes aufgeführt. Angaben über Preise, Zusatzleistungen, Bilder, Beschreibungstexte, Vakanzen, Rabatte und Buchungshinweise liegen in voneinander getrennten Dateien vor. Aus ihnen lässt sich das Gerüst für die zukünftige Datenbank bilden. Es kann durchaus möglich sein, dass Teile der Daten nur über eine Schnittstelle des Anbieters abrufbar sind. Sie müssen dann direkt bei Aufruf der entsprechenden Objekte abgeholt und in die Ausgabe integriert werden.

Die Auswertung bildet die Grundlage für folgende Tabellenstruktur:

- Stammdaten
- Merkmale
- Beschreibungstexte Haus
- Beschreibungstexte Wohnung
- Bilder Haus
- Bilder Wohnung
- Rabatte
- Zusatzleistungen
- Preise
- Buchungshinweise
- Vakanzen

Die Daten sind dabei durch den Anbietercode zu ergänzen, welcher vorher für jeden Anbieter eindeutig festzulegen ist. Für eine spätere Auswertung ist dieser neue Code zwingend nötig!

3.4.2. Erzeugung eigener Objektcodierung

Jeder Anbieter bringt im übrigen seine ganz eigene Codierung der Länder, Regionen und Orte mit. An dieser Stelle ist eine eigene Codierung vorgesehen. Einträge, welche in die Datenbank einfließen, müssen also Konvertiert werden. Es ist davon auszugehen, dass dieser Prozess kein einfaches Unterfangen wird, da die Codierung als Gemeinsamkeit herausfällt. Die Konzentration liegt auf dem Namen an für sich. Für die eigene Kodierung sind drei weitere Tabellen nötig. Dies sind:

- Codierung Land
- Codierung Region

- Codierung Ort

Über einen Algorithmus soll dann eine Verbindung zwischen den Codierungen der Anbieter und der internen Codierung hergestellt werden. Erst wenn diese Verbindung besteht ist es möglich, eine Konvertierung der Stammdaten durchzuführen. Angedacht ist dabei der Aufbau einer Referenztafel für jeden Anbieter in der jede Codierung mit der internen Codierung verknüpft wird. Da sich ständig Änderungen in den Stammdaten bezüglich Anzahl der Objekte, Wegfall und Aufnahme neuer Länder, Regionen und Orte ergeben, ist vor dem Aktualisieren der Stammdaten ein Update auf die Referenztafel des Anbieters unumgänglich. Aufgrund des Datenumfanges kann dies auch nur automatisiert erfolgen! Die Datenbank erhält also folgende zusätzliche Tabellen

- Referenzierung Interhome
- Referenzierung Wolters
- Referenzierung altes zu neuem System

Der letzte Punkt ergibt sich aus der Tatsache, dass auch Einträge des aktuellen Systems in das neue übernommen werden. Dabei ist ebenfalls eine Konvertierung nötig.

Einige Zeit nach Etablierung dieser Art von Referenzierung kam die Idee auf, von den Referenztabellen der Anbieter abzugehen und die Referenzierung in bestehende Referenztabellen von Ländern, Regionen und Orten des Systems zu integrieren. Dabei werden die Tabellen durch weitere Spalten ergänzt, in denen dann die Codierung, wie sie der Anbieter verwendet, hinterlegt ist. Aktuell ist das nur eine Idee, die noch nicht in die Tat umgesetzt ist. Eine Änderung könnte jedoch in Zukunft erfolgen, da sich Redundanzen abbauen lassen und sich der Prozess der Datenkonvertierung eventuell sogar vereinfachen lässt.

3.4.3. Zusammenfassung von Mehrdeutigkeiten

Zu berücksichtigen ist die Tatsache, dass geografische Bezeichnungen, die das selbe Gebiet definieren, bei den einzelnen Anbietern unterschiedlich bezeichnet sein können. In der Suchmaske wird sich dadurch das Problem ergeben, dass es mehrere Auswahlmöglichkeiten für ein und dasselbe Gebiet gibt und alle Möglichkeiten höchstwahrscheinlich unterschiedliche Objekte ausgeben. Infolgedessen ist es ungünstig die Referenztabellen der Länder, Regionen und Orte als Grundlage für ein Auswahlménü in der Suchmaske zu nehmen. Es muss eine Lösung geschaffen werden die Einträge vereint. Denkbar wären hier z.B. separate Tabellen:

- Menü Land
- Menü Region
- Menü Ort

Außerdem eine zusätzlich Tabelle, die Verknüpfungen zwischen den sich ähnelnden Einträgen herstellt.

- Geografische Schnittmenge

Problematisch ist dabei einen Anhaltspunkt festzumachen durch den genau definierbar ist, welche Gebiete zusammen gehören und welche nicht. Hier lässt sich ein manueller Eingriff u.U. nicht vermeiden!

3.4.4. Speicherung von Vorgängen

Jeder Buchungsvorgang muss aus Sicherheitsgründen für einen gewissen Zeitraum erfasst werden. Eventuelle Komplikationen und damit verbundenen Rückfragen seitens der Personen, welche gebucht haben, lassen sich besser zurück verfolgen. Gleichzeitig existiert damit ein schriftlicher Beweis mit Angabe des Datums an dem die Buchung stattgefunden hat und einer Angabe des Buchungsstatus. Es folgt also eine weitere Tabelle:

- Buchungsvorgang

3.5. Aggregation

Anbieterdaten müssen in den Datenpool integriert und stets auf dem neuesten Stand gehalten werden. Es ist davon auszugehen, dass gelieferte Datensätze nicht 1:1 in die Datenbank übertragbar sind. Deshalb muss im Vorfeld eine Anpassung an das System erfolgen. Daneben sind weitere Bedingungen zu berücksichtigen:

- Automatisierte Beschaffung und ggf. Extraktion der Daten
- Konvertierung
- Erzeugung einer neuen Datei in der die Datensätze in der Struktur der entsprechenden Tabelle des Systems abgespeichert werden
- Löschen aller Datensätze des Anbieters in entsprechender Tabelle
- Laden der neuen Datensätze des Anbieters in die Tabelle
- Protokollierung der Aktionen
- Sofortiger Abbruch bei kritischem Fehler
- Im Falle eines kritischen Fehlers Admin per E-Mail kontaktieren

Der Aufruf der Scripte, welche für die Aktualisierung zuständig sind, soll über eine sog. Cron Steuerung erfolgen. Dabei ist zu berücksichtigen, dass es je nach Daten unterschiedliche Zyklen der Aktualisierung gibt. Aus der Dokumentation der Anbieter lässt sich hierbei folgendes Schema ableiten:

- Aktualisierung von Vakanzen 1 - 2 mal täglich
- Aktualisierung der Objektdaten 1 - 2 mal wöchentlich

3.6. Kommunikation und Export

Da die Anbieter über sog. Webservices verfügen und diese für den Bezug aktueller Daten angesprochen werden müssen, sind noch folgende Punkte zu beachten:

- Art der Schnittstelle ermitteln
- Für die Kommunikation mit der Schnittstelle erforderliche Maßnahmen treffen
- Gegenkontrolle bestimmter Werte aus der Datenbank über die Schnittstelle
- Werte sind Beispielsweise Vakanzdaten, Preise
- Versand von Informationen an die Schnittstelle, z.B. Buchungsdaten
- Auswertung von Rückgabewerten, z.B. bei einer Buchung

Daten aus dem bestehenden Ferienhaus System müssen in das neue System übernommen werden. Es ist ein Export aller Daten erforderlich. Dies umfasst:

- Sammeln der Daten
- Konvertierung von Daten, z.B. Objektcodierung, Vakanzen
- Zusammenfassung von Daten, z.B. Merkmale
- Strukturierung nach den Vorgaben des neuen Systems
- Aufbereitung als versandfertige Dateien

4. Vergleich von mehreren, stichprobenartig ausgewählten Anbietern

4.1. Allgemeines Erscheinungsbild

Ausgewählte Anbieter sind *e-domizil.de*, *interchalet.com*, *tourist-online.de*, *ferienhaus.com*.

Die Aufteilung der Seiten ist bei den Anbietern gleich. Wird das Portal betreten steht die Suchmaske im Mittelpunkt. Entweder muss im ersten Schritt ein Land ausgewählt werden¹² oder der Besucher hat sofort die Möglichkeit Land, Region, Ort, Reisedatum und Merkmale auszuwählen.¹³ Zudem besteht in den meisten Suchmasken die Möglichkeit, zwischen einer einfachen oder erweiterten Suche zu wählen. Angebote zu bestimmten Themen, Last Minute-Angebote und Serviceinformationen umrahmen die Suchmaske.

Der Suchmaske folgt eine Übersicht der den Kriterien entsprechenden Angebote. Jeder Anbieter listet zum Objekt ein Vorschaubild, Kurzbeschreibung, Angabe wichtiger Merkmale, Lagebeschreibung und einen Preis. Dabei handelt es sich um einen „ab“ Preis oder um einen Saisonpreis. Teilweise listen Anbieter auch eine Preisspanne vom niedrigsten bis zum höchsten Preis. Pro Seite sind in vielen Fällen zehn Objekte aufgeführt. Eine Änderung der Anzahl von Objekten, die pro Seite ausgegeben werden, ist bei allen ausgewählten Anbietern nicht möglich.

Die Detailansicht offenbart von Anbieter zu Anbieter Unterschiede. Allen ist eine Bildergalerie und die ausführliche Beschreibung des Objektes gemein. Eine Auflistung von Merkmalen findet schon nicht mehr bei jedem Anbieter den Weg in die Detailansicht. Nur knapp die Hälfte bietet eine Preis- und Belegungsübersicht. Dafür legen andere Anbieter wiederum Wert auf eine ausführliche Merkmals- und Themenaufstellung. Tourist Online, Interchalet und e-domizil stellen ihre Seiten sogar in mehreren Sprachen zur Verfügung.

Eine Buchung findet je nach Anbieter auf anderem Weg statt. Zwei Möglichkeiten stehen zur Auswahl. Bei einem Teil der Anbieter ist es möglich direkt auf der Webseite zu buchen. Es gibt jedoch Anbieter, wo der Besucher Kontakt zum Vermieter aufnehmen muss. Telefonnummer und Kontaktadresse sind hinterlegt.

¹² <http://www.interchalet.com> / <http://www.tourist-online.de>

¹³ <http://www.e-domizil.de> / <http://www.ferienhaus.com>

Erfolgt eine Buchung auf dem Portal des Anbieters, verlangt dieser einige Informationen bevor die eigentliche Buchung ausgelöst wird. Die buchungswillige Person muss Angaben über Anreisedatum, Aufenthaltsdauer, Anzahl reisender Personen und Reiseversicherung machen. Persönliche Angaben der buchenden Person sind ebenso zu treffen. Zu den weiteren Schritten gehört auch die Auswahl der Zahlungsart. Erst wenn alle Angaben getätigt wurden und der User sie auf einer Übersichtsseite bestätigt hat, wird die eigentliche Buchung ausgelöst.

4.2. Suchmaske der Anbieter

E-domizil bietet drei Varianten der Suche. Neben einer Standard- und erweiterter Suche ist es dem Besucher möglich auch eine Volltextsuche in Anspruch zu nehmen. Ei-

FERIENWOHNUNGEN UND FERIENHÄUSER SUCHEN

Land (Objekte)
Bitte wählen Sie

Region (Objekte)
- alle -

Anzahl Personen
- egal -

Anreiseternin
Ich möchte genau am gewählten Termin anreisen.

Reisedauer
1 Woche

Objekttyp
- alle -

Ausstattung:

<input type="checkbox"/> Waschmaschine	<input type="checkbox"/> Swimmingpool	<input type="checkbox"/> TV
<input type="checkbox"/> Geschirrspüler	<input type="checkbox"/> Sauna	<input type="checkbox"/> Haustier erlaubt

Suchen

[Suche zurücksetzen](#) [Erweiterte Suche](#) [Volltextsuche](#)

Abbildung 4.1.: Standardsuche, e-domizil.de¹⁴

ne Standardsuche beschränkt sich auf wesentliche Auswahlmöglichkeiten. Dazu zählen Auswahl des Zielgebietes, Anzahl reisender Personen, gewünschter Anreiseternin, Aufenthaltsdauer, Objekttyp und eine kleine Auswahl von häufig gefragten Merkmalen. Ist der reisewilligen Person bereits klar, wo sie hinreisen will, besteht die Möglichkeit einer Volltextsuche. Der Reiseort wird einfach eingetippt und nach ca. 1 - 2 Sekunden erscheint eine Auswahlliste mit passenden Ergebnissen. Die Möglichkeit Ausstattungsmerkmale und Objekttyp zu wählen besteht hier nicht. Die erweiterte Suche bietet ein

¹⁴ <http://www.e-domizil.de> - Option „Schnellsuche“ in der Buchungsmaske

¹⁵ <http://www.e-domizil.de> - Option „Volltextsuche“ in der Buchungsmaske

¹⁶ <http://www.e-domizil.de> - Option „Mehr Optionen“ in der Buchungsmaske

Erweiterte Suche nach einem Ferienhaus und Ferienwohnung

Hier haben Sie die Möglichkeit zu einer erweiterten Suche nach einer Ferienwohnung oder einem Ferienhaus. Es stehen Ihnen viele verschiedene Auswahlmöglichkeiten zur Verfügung, so dass Sie die Suchergebnisse präzise auf Ihre Wünsche abstimmen können.

Suchbegriff(e): z.B.: Mallorca Finca Pool ⓘ

Anreiseternin: ⓘ

☐ Ich möchte genau am gewählten Termin anreisen. ⓘ

Anzahl Personen: ⓘ

Reisedauer: ⓘ

Suchen

Abbildung 4.2.: Volltextsuche, e-domizil.de¹⁵

Erweiterte Suche nach einem Ferienhaus und Ferienwohnung

Hier haben Sie die Möglichkeit zu einer erweiterten Suche nach einer Ferienwohnung oder einem Ferienhaus. Es stehen Ihnen viele verschiedene Auswahlmöglichkeiten zur Verfügung, so dass Sie die Suchergebnisse präzise auf Ihre Wünsche abstimmen können.

Land (Objekte): ⓘ

Region (Objekte): ⓘ

Ort: ⓘ

Anzahl Personen: ⓘ

Anzahl der Schlafzimmer: ⓘ

Objekttyp: ⓘ

Angebot: ⓘ

Preisspanne von: EUR/Woche

bis: EUR/Woche

Anreiseternin: ⓘ

Reisedauer: ⓘ

☐ Ich möchte genau am gewählten Termin anreisen. ⓘ

Entfernungen:

Meer/ See: ⓘ

Skilift/ Bergbahn/ Loipe: ⓘ

Ausstattung:

☐ Waschmaschine ☐ Geschirrspüler ☐ TV

☐ Swimmingpool ☐ Sauna ☐ Haustier erlaubt

Suchen

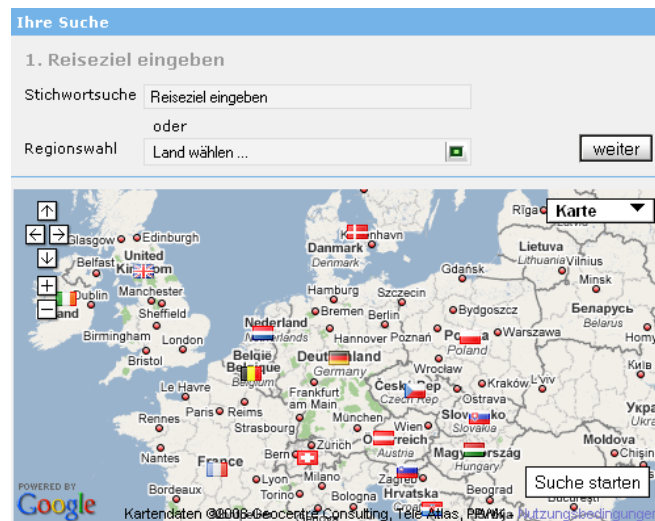
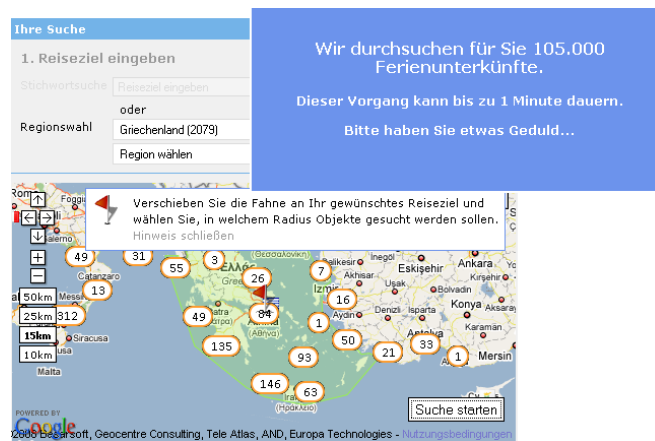
Abbildung 4.3.: Erweiterte Suche, e-domizil.de¹⁶

paar zusätzliche Auswahlkriterien. Dazu gehören Bestimmung einer Preisspanne, Ortsauswahl, Entfernungen zu Skianlagen oder Meer, Schlafzimmeranzahl und Angebotsart.

Tourist-Online legt den Schwerpunkt auf eine Suche per Karte. Es gibt die Möglichkeit sein Reiseziel über die Stichwortsuche zu finden oder eine Region auszuwählen. Das Land ist ebenfalls über die Karte einzugrenzen. Gezeigt ist eine Europakarte. Alle Länder, in denen Ferienobjekte liegen, sind mit einer Landesflagge gekennzeichnet. Ist die Auswahl auf ein Land gefallen, zoomt die Karte in das Land hinein. Jede Region des Landes enthält nun einen Bereich, der anzeigt wie viele Objekte in dieser Region verfügbar sind. Diese Punkte sind anwählbar und leiten den User auf die Übersichtssei-

¹⁷ <http://www.tourist-online.de>

¹⁸ <http://www.tourist-online.de>

Abbildung 4.4.: Suchmaske, tourist-online.de¹⁷Abbildung 4.5.: Regionssuche, tourist-online.de¹⁸

te, wo eine Ausgabe aller Objekte der Region erfolgt. Die Auswahl eines Reisedatums, Ausstattungsmerkmale und Anzahl Reisender ist in dieser Art der Suche nicht integriert.

Ein Link am unteren Ende der Standardsuche klappt den Bereich für die erweiterte Suche auf. Es besteht nun die Möglichkeit Lage, Haustyp, Zimmeranzahl und einige Ausstattungsmerkmale als Kriterien in die Suche einfließen zu lassen.

Ferienhaus.com ordnet die Suche als Menü im linken Bereich der Seite ein und lässt den Contentbereich für einen Einleitungstext frei. Die Suchmaske bietet dabei vergleichbare Optionen wie bereits genannte Anbieter in der Standardsuche. Ausstattungsmerkmale sind nicht wählbar. Allerdings ist die Eingabe einer Objektzahl möglich. So lassen sich einzelne Objekte direkt wiederfinden.

¹⁹ <http://www.interchalet.com>

Ferienhaus suchen / Dänemark

Region: -keine Einschränkung-

Ort: -keine Einschränkung-

Anzahl Personen: -keine Einschränkung-

Haustier reist mit: ☐ ja

Anreisedatum:

Aufenthaltsdauer: -keine Einschränkung-

Erweiterte Suche

Zimmerzahl: -keine Einschränkung-

Lage: -keine Einschränkung-

Haustyp: -keine Einschränkung-

Ausstattung:

<input type="checkbox"/> Pool	<input type="checkbox"/> Sauna
<input type="checkbox"/> Tennis	<input type="checkbox"/> TV
<input type="checkbox"/> Geschirrspüler	<input type="checkbox"/> Waschmaschine
<input type="checkbox"/> Heizung	<input type="checkbox"/> Garage

[Alle Eingaben löschen](#) [Angebote anzeigen](#)

Abbildung 4.6.: Suchmaske erweiterte Auswahl, interchalet.com¹⁹

4.3. Objektübersicht

E-domizil listet Objekte in einer fünf Spalten Ansicht. Im Kopf der Übersicht zeigt der Anbieter die Anzahl der Treffer, aktuelle Seite und die geografische Lage der Objekte an.

Ein Link bietet die Möglichkeit, eine neue Suche zu starten. Eine Sortierung der Ergebnisse ist möglich. Pro Seite listet E-domizil zehn Ergebnisse auf. Aufgezählt von links, findet sich in der ersten Spalte ein Vorschaubild und Piktogramme wieder. Piktogramme stellen Merkmale des Objektes dar. Es folgt in der nächsten Spalte eine Kurzbeschreibung zum Objekt. Die dritte Spalte gibt Aufschluss über Lage und Objekttyp. Spalte vier listet die Anzahl maximal erlaubter Personen pro Objekt. Die letzte Spalte enthält Angaben über den Preis und bietet Links zur Detailansicht oder einen Eintrag in die Merkliste. Hervorzuheben ist die Möglichkeit nach Preis, Anzahl Personen, Ort, Region und Objektname zu sortieren. Eine Sortiermöglichkeit wird nicht von jedem Anbieter geboten.

²⁰ <http://www.e-domizil.de> - Bei erfolgter Suche nach Objekten

Suchergebnisse: **Treffer 27** | Seite **1 von 3**

Gesucht wurde nach...

Land: **Spanien** | Region: **Ibiza**

Suche verändern

Sortieren nach:

Empfehlung

Bild	Objektname / Kurzbeschreibung	Ort	Region	Personen	Preis/Woche
	IBIZA/CALA VADELLA E57430.100.1 Haus: Haus, ausserhalb des Ortes, 1,5 km vom Zentrum von Cala Vedella, erhöhte Lage. Zur Alleinbenutzung: Grundstück 1'500 m2 (eingezäunt), Garten 200 m2 mit Pflanzen und Bäumen, Freibad (6 x 9 m, 01.04.-30.10.), Aussendusche, Terrasse (140 ...	IBIZA/CALA VADELLA	Ibiza	6 Pers.	ab 1.036,00 EUR* Ansehen Merken
	HAUS 2 FINCA ROMANTICA CALA COMTE IBIZA Romantische Finca auf 1300 m2 gepflegter Gartenanlage in Naturschutzgebiet. Die 3 Fincas liegen auf abgeschlossenem Grundstück mit Swimmingpool und Garten, von der Straße nicht einsehbar im Grüngürtel mit ländlichem Ambiente der Cala Co ...	SAN JOSE IBIZA	Ibiza	2 - 4 Pers.	ab 550,00 EUR* Ansehen Merken

Abbildung 4.7.: Objektübersicht, e-domizil.de²⁰

Tourist-Online bietet auf der Übersicht im Kopfbereich der Seite die Art von Suchmaske, wie sie die vorgenannten Anbieter bereits auf ihrer Startseite vorweisen. Sie ermöglicht eine genauere Suche. In der Vorauswahl sind jedoch schon viele Objekte gelistet. Die Daten zum Objekt werden in eine Zwei-Spalten Ansicht aufgeteilt. In der linken

Umkreisuche 15km bei Griechenland, Athen und Umiland, Aghios Dimitrios

Reisziel eingeben oder

Griechenland (2079) ☒ Tag ☒ Monat der Anreise ☒ Ferienhäuser und Ferienwoh.


Athen und Umiland (205) ☒ Mo Di Mi Do Fr Sa So 2 Personen ☒

alle Orte ☒ 1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

7 Nächte (1 Woche) ☒
max. Preis wählen ☒
Sortierung wählen ☒

Erweiterte Suche >

1 - 4 von 4 Ergebnissen



"ASTERIA" für 6 Personen in Aghios Dimitrios, Südpeloponnes (Griechenland) [983091]


Griechenland Peloponnes Aghios Dimitrios

1 km außerhalb des Ortes Aghios Dimitrios gelegen, mit weitem, schönen Meerblick steht das Haus Asteria mit insgesamt 4 Wohnungen. Direkt über dem Meer mit weitem, phantastischem Blick auf die Messinische Bucht, Sonnenuntergangseite, liegt das gepflegte Anwesen ASTERIA bestehend aus 2 Häusern. ... [\[mehr\]](#)

Objekt Nr.: 983091 Anreisetag: Täglich

+5 Fotos Wochenpreis: 525,00 EUR - 945,00 EUR

Entfernung zur gewählten Position: ca. 0 km ¹



Südküste Kreta: Privates Ferienhaus auf Kreta - idyllisch, einzigartig [805813]

Griechenland Kreta Mittelmeerregion Attika Griechische Inseln Griechisches Festland Athen und Umiland Südeuropa Tyrmbaki Festos Agios Ioannis

Das Haus auf Kreta ist unser privates Ferienhaus. Da wir voll berufstätig sind und daher leider nicht so oft und so lange dort sein können, vermieten wir es in der übrigen Zeit gerne an Sie. Wir haben das Haus mit sehr viel Liebe zum Detail für unsere eigenen - recht hohen - Ansprüche ... [\[mehr\]](#)

Objekt Nr.: 805813 Preis: siehe Detail Beschreibung

+6 Fotos Entfernung zur gewählten Position: ca. 3 km ¹

☎ 07071-9425914

Abbildung 4.8.: Objektübersicht, Tourist-online.de²¹

Spalte befindet sich ein Vorschaubild, die Objektnummer und ein Link zur Fotogalerie. Die rechte Spalte enthält den Objektamen, Lageangabe und eine Kurzbeschreibung, sowie Preise, Anreisezeitraum und Darstellung wichtiger Merkmale als Piktogramm. Über die darunter aufgelisteten Buttons ist ein Aufruf von Detailbeschreibung, Karte, ein Eintrag in die Merkliste und Anfrage als auch Buchung möglich. Tourist-Online listet pro Seite 8 Objekte.

²¹ <http://www.tourist-online.de> - Bei erfolgter Suche nach Land, Region, Ort

Interchalet gestaltet die Übersicht kompakt. Im Kopf befinden sich Angaben zur geografischen Lage der Objekte und ein Verweis auf die Suchmaske um die Auswahlkriterien zu ändern. Eine darunter befindliche Navigationsleiste zeigt dem User an wie viele Treffer zur Auswahl existieren und auf welcher Seite er sich befindet. Zusätzlich bekommt

Meine Auswahlkriterien
Dänemark , Nordsee: Westjütland
[Auswahlkriterien ändern](#)

Trefferliste: 17 Objekte [Drucken](#) Seite 1 / 4

Sortieren nach: Preis: Personenzahl: Ort:

Sauna TV Heizung	Ferienhaus mit Sauna, Houstrup (Nordsee: Westjütland), Dänemark Belegung: bis 5 Personen Preis pro Woche ab: € 152	HST 122 weiter
Karte / Wetter <input type="checkbox"/> Zum Merkzettel hinzufügen		
TV Waschmaschine Heizung Kinderbett	Ferienhaus mit Sauna, Houstrup (Nordsee: Westjütland), Dänemark Belegung: bis 6 Personen Preis pro Woche ab: € 224	HST 128

Abbildung 4.9.: Übersicht, interchalet.com²²

er die Möglichkeit nach Preis, Personenanzahl und Ort zu sortieren. Interchalet teilt die Übersicht in drei Spalten auf. In der linken Spalte sind Merkmale aufgeführt. Die mittlere Spalte gibt Auskunft über wichtige Merkmale und Lage, maximale Belegung und einen Wochenpreis. Als Service ist nicht nur der Link zur Karte, sondern auch die Wetterlage der Region über einen Link abrufbar. Die Auswahl „Zum Merkzettel hinzufügen“ schließt die mittlere Spalte ab. Ein Vorschaubild, Objektcode und der Link zur Detailansicht befinden sich in der rechten Spalte.

Ferienhaus.com weicht bei der Objektdarstellung in der Übersicht vom bekannten Schema ab. Auffällig ist die Formatierung der Einträge. Ungerade und gerade Einträge bekommen unterschiedliche Hintergrundfarben. Dadurch fällt ihre Unterscheidung und das Lesen der Liste leichter. Sie lassen sich nach bestimmten Kriterien auf- und absteigend sortieren. Aufgeteilt in drei Spalten gibt es zusätzlich eine Unterteilung in zwei Zeilen. Jeder Eintrag führt im Kopf die geografische Lage, Objekttyp und Code. Die erste Zeile enthält links ein Vorschaubild, mittig Typangabe, Lage, Mindestpreis und Preis für den gewählten Zeitraum. Rechts sind Merkmale wie maximale Belegung, Anzahl von Schlafzimmern und Bädern angegeben. Die zweite Zeile bietet in der linken Spalte

²² <http://www.interchalet.com> - bei erfolgreicher Suche und der Auswahl „Anzeigen“

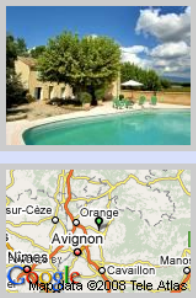
²³ <http://www.ferienhaus.com> - nach erfolgter Suche

Objekt

41 Objekte von insgesamt 41 gefunden
Seite auswählen: [1] 2 3

Sortieren nach Sortiermethode

■ Frankreich - Provence - Avignon Umgebung - Lorient du Comtat - Ferienhaus - PRV087



Typ: Ferienhaus
Lage: Lorient du Comtat / Avignon Umgebung / Frankreich - Provence

Preis ab: 1.543 EUR/Woche
14. Mär. 2009 - 21. Mär. 2009: **1543,00 EUR**

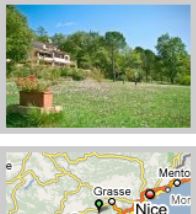
Summe (ohne Nebenkosten) (7Nächte) 1543 EUR

Personen: 10
Schlafzimmer: 4
Bäder: 2
Swimmingpool: Ja

Dieses Mas bei Lorient ist ein ursprüngliches provenzalisches Landhaus aus dem Jahre 1883 und wurde 2005 komplett renoviert, ohne seinen Charme zu verlieren. Nun finden Sie ein äußerst komfortables Ferienhaus vor mit großen Schlafräumen, neuen Bädern und provenzalischem Flair.

[Zur Merkliste](#) [Details ansehen](#)

■ Frankreich - Côte d'Azur - Cannes Umgebung - Montauroux - Ferienwohnung - COT051



Typ: Ferienwohnung
Lage: Montauroux / Cannes Umgebung / Frankreich - Côte d'Azur

Preis ab: 975 EUR/Woche
14. Mär. 2009 - 21. Mär. 2009: **975,00 EUR**

Summe (ohne Nebenkosten) (7Nächte) 975 EUR

Personen: 8
Schlafzimmer: 4
Bäder: 3
Swimmingpool: Ja

In den Hügeln oberhalb von Cannes, direkt am Lac de St. Cassien

Abbildung 4.10.: Übersicht, ferienhaus.com²³

te eine geografische Karte von Google²⁴ mit genauer Standortangabe. Mittig steht die Kurzbeschreibung. Im Fußbereich jedes Objektes findet sich die Möglichkeit das Objekt vorzumerken oder die Detailansicht aufzurufen.

4.4. Detailansicht

E-domizil unterteilt die Seite in eine tabellarische Ausgabe wichtiger Daten, Bildergalerie, Beschreibungen von Haus und den Zimmern und einer ausführliche Ausstattungsbeschreibung. Die Auflistung von Ausstattungsmerkmalen bezieht sich auf einzelne Räumlichkeiten, Aussenanlagen und die Umgebung. In der Detailansicht können Preise und die Verfügbarkeit abgefragt werden. Unter Verfügbarkeit findet sich ein Kalender. Er bietet eine Belegungsvorschau von bis zu zwei Jahren. Für die Urlaubsplanung stellt e-domizil eine Option bereit, um auch die Schulferien für ein bestimmtes Bundesland in der Anzeige zu berücksichtigen. Sind Anreisetermin, Reisedauer und Personenanzahl gewählt, ist eine Abfrage von Preisen und Verfügbarkeit für diesen Zeitraum möglich. Für eine schnelle Übersicht gibt e-domizil bereits Preise auf Wochenbasis für einzelne Saison- und Monatsabschnitte an. Eine Karte zur visuellen Beschreibung der Lage ist

²⁴ <http://www.google.com>

²⁵ <http://www.e-domizil.de> - Auswahl eines Objektes in der Übersicht

²⁶ <http://www.e-domizil.de> - Auswahl Karteikarte in der Detailansicht

OBJEKTbeschreibung	PREISE UND VERFÜGBARKEIT	KARTE	BUCHUNG														
Ferienhaus Vrsar-Marasi CIL351 - Objektnummer: 288601																	
Land:	Kroatien	Prüfen Sie Preise und Verfügbarkeit Ihres gewünschten Feriendomizils durch Angabe des Anreisetages, der Reisedauer und der Personenzahl. Hier prüfen															
Region:	Istrien																
Ort:	MASARI																
Ausstattung:																	
Bewertung:	Keine Bewertung vorhanden																
Objekttyp:	Ferienhaus																
Belegung:	6 Personen																
Zurück zu den Suchergebnissen		Zur Buchung Zur Buchung															
Bilder im Großformat anzeigen Diashow Send2Friend Merken Merken Drucken																	
<p>MASARI Vrsar-Marasi CIL351 Ferienhaus in MASARI, 110 m² Wohnfläche, 350 m² Grundstück, 6 Personen, 3 Schlafzimmer, 3 Bäder, Offene und überdachte Terrasse, Aussenpool, Whirlpool, 3-TV, SAT-Empfang, Geschirrspüler, Waschmaschine. Entfernung Felsenküste: 8000 m. Entfernung Einkaufsmöglichkeit: 2500 m. Entfernung Restaurant: 1000 m. Entfernung Stadt: 7000 m. 2 Doppelbett(en), 2 Etagenbett(en) sind im Objekt vorhanden. Nähere Informationen können Sie ggf. dem Grundriss entnehmen</p> <p>Zimmerbeschreibung. Saison 2008: Ferienhaus Schlafzimmer: 2 Schlafplätze(Doppelbett) Schlafzimmer 2: 2 Schlafplätze(Doppelbett) Schlafzimmer 3: 2 Schlafplätze(Etagenbett) Wohnraum/Küche: 2 Schlafplätze(Schlafsofa) Badezimmer: WC, Warmes und kaltes Wasser, Dusche und Badewanne. Badezimmer 2: WC, Warmes und kaltes Wasser, Badewanne. Badezimmer 3: WC, Warmes und kaltes Wasser, Dusche. Terrasse: Offene und überdachte Terrasse. Im Reisepreis ist eine Reiserücktrittsversicherung enthalten.</p> <p>Nebenkosten: Stromkosten inkl. ganzjährig Saison 2009: Bettwäsche: 0,00 EUR pro Garnitur (optional), gültig von 20.09.2008 bis 10.01.2009; Endreinigung: 0,00 EUR pro Objekt (optional), gültig von 20.09.2008 bis 10.01.2009; Saison 2009: Bettwäsche: 0,00 EUR pro Garnitur (optional), gültig von 10.01.2009 bis 09.01.2010; Endreinigung: 0,00 EUR pro Objekt (optional), gültig von 10.01.2009 bis 09.01.2010;</p>																	
<p>Ausstattung Zu beachten: Weitere Objektinformationen entnehmen Sie bitte dem ausführlichen Beschreibungstext.</p> <p>Kurzbeschreibung</p> <table border="1"> <tbody> <tr> <td>Art des Objektes:</td> <td>Ferienhaus</td> </tr> <tr> <td>Anreisetag:</td> <td>Samstag</td> </tr> <tr> <td>Wohnfläche:</td> <td>110 m²</td> </tr> <tr> <td>Grundstücksfläche:</td> <td>350 m²</td> </tr> <tr> <td>Normalbelegung:</td> <td>6 Personen</td> </tr> <tr> <td>Maximalbelegung:</td> <td>6 Personen</td> </tr> <tr> <td>Haustier:</td> <td>erlaubt</td> </tr> </tbody> </table>				Art des Objektes:	Ferienhaus	Anreisetag:	Samstag	Wohnfläche:	110 m²	Grundstücksfläche:	350 m²	Normalbelegung:	6 Personen	Maximalbelegung:	6 Personen	Haustier:	erlaubt
Art des Objektes:	Ferienhaus																
Anreisetag:	Samstag																
Wohnfläche:	110 m²																
Grundstücksfläche:	350 m²																
Normalbelegung:	6 Personen																
Maximalbelegung:	6 Personen																
Haustier:	erlaubt																

Abbildung 4.11.: Detailansicht Objekt, e-domizil.de²⁵

ebenfalls abrufbar. Sie basiert auf dem Kartensystem von Google Maps(FUßNOTE). Dadurch sind von Google bekannte Darstellungen wie Karten, Satellitenbilder oder eine Mischung aus beiden Varianten möglich. Eine ungefähre Positionsermittlung ist durch die Postleitzahl des Ortes möglich. Daneben kann sofort gebucht werden. Das System leitet den User auf den Buchungsvorgang, wo er alle nötigen Angaben tätigt.

Tourist-Online bietet eine Bildvorschau. Wird der Mauszeiger über ein Vorschaubild bewegt, öffnet es sich auf einem Layer²⁷ und verschwindet wieder wenn das Vorschaubild verlassen wird. Erfolgt ein Klick öffnet sich die Galerie. Eine Beschreibung gehört ebenso zur Detailansicht. Der Anbieter stellt sie wahlweise in vier unterschiedlichen Sprachen zur Verfügung. Es folgt eine Auflistung der Ausstattungsmerkmale. Sie gibt auch Auf-

²⁷ Skop 2007, Stylesheets (CSS) - CSS-Eigenschaften - Positionierung und Anzeige von Elementen

²⁸ <http://www.tourist-online.de> - Auswahl eines Objektes aus der Übersicht

Verfügbarkeit des Objektes

Januar 2009

Mo	Di	Mi	Do	Fr	Sa	So
			01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Ferien: Berlin

Legende:

20 Anreisetag 20 Frei
20 Belegt 20 Schulferien

Anreisetag: * 10 01 2009

Reisedauer: 1 Woche

Anzahl Personen: 2

* Bitte wählen Sie das Datum mit Hilfe des Kalenders.

[Preise und Verfügbarkeit anzeigen](#)

Preise		
Hauptsaison	18.07.2009 - 22.08.2009	1.799,00 EUR / Woche*
Zwischensaison	04.07.2009 - 18.07.2009	1.620,00 EUR / Woche*
Nebensaison	20.09.2008 - 10.01.2009	739,00 EUR / Woche*
	10.01.2009 - 23.05.2009	830,00 EUR / Woche*
	23.05.2009 - 20.06.2009	940,00 EUR / Woche*
	20.06.2009 - 04.07.2009	1.190,00 EUR / Woche*
	22.08.2009 - 05.09.2009	1.190,00 EUR / Woche*
	05.09.2009 - 19.09.2009	940,00 EUR / Woche*
	19.09.2009 - 09.01.2010	830,00 EUR / Woche*

* Nutzen Sie bitte nach Eingabe des Anreisetages, der Reisedauer und der Personenzahl die Funktion "Preis und Verfügbarkeit anzeigen", um den genauen Reisepreis zu errechnen.

[Seitenanfang](#) [Zur Buchung](#)

Abbildung 4.12.: Detailansicht Kalender, e-domizil.de²⁶

schluss, ob eine Reiserücktrittskostenversicherung vorliegt. Die Ausstattungsmerkmale stehen in den Sprachen Deutsch und Englisch zur Verfügung. Die Belegungsübersicht zeigt bis zu ein Jahr im voraus den Status des Objektes an. Ein Objekt kann zu einem Zeitpunkt frei, belegt oder nur auf Anfrage buchbar sein. Die anschließende Preisübersicht gibt dem Reisewilligen Auskunft über die Saisonpreise, mögliche Anreisetage und maximale Anzahl an Personen, die im Objekt untergebracht werden können. Zusätzliche Informationen geben u.a. Aufschluss über Kautionspreis, Preis einer Reinigung und sonstige Nebenkosten. Wahlweise ist eine Anzeige in Deutsch oder Englisch möglich. Über die Navigationsleiste am Beginn und Ende der Detailansicht ist eine Buchung, Anfrage, Reservierung oder Eintragung in die Merkliste möglich. Die Navigation am rechten Bildrand ermöglicht eine geografische Anzeige des Objektes auf Karten von Google.

Interchalet gestaltet die Detailansicht gegenüber den vorgenannten Anbietern nicht so umfangreich. Die Kopfzeile enthält ein Vorschaubild, geografische Lage, Objektcode, maximal mögliche Belegung und einen Mindestpreis. Darunter folgt eine Aufteilung in drei Spalten. Im Mittelpunkt steht eine ausführliche Objektbeschreibung. Sie enthält Hinweise über Kautionspreis, Ausstattung, Zusatzleistungen und Nebenkosten. Jedes Zimmer im Objekt wird einzeln beschrieben. Eine Preistabelle listet die Preise nach Saison auf. Eine Belegungsübersicht zeigt die Belegung des aktuellen Monats an.

Die Spalten auf der linken und rechten Seite bieten die Möglichkeit einen Anreise-

²⁹ <http://www.interchalet.com> - Auswahl eines Objektes in der Übersicht

Ferienwohnung - Erdgeschoss RESIDENCIA PANCO in Rojales bei Torrevieja, Costa Blanca, Ciudad Quesada für 4 Personen (Spanien) [352177]

[Spanien](#) [Murcia](#) [Valencia](#) [Costa Blanca](#) [Rojales](#) [Torrevieja](#) [Ciudad Quesada](#)



[Bildergalerie ansehen](#)

Spanien, Rojales, max. 4 Personen, Wohngröße: 64m². Gemeinsames Grundstück, Erdgeschoss, Stromheizung transportabel, Winterfest, Balkon und Terrasse, Miniküche; w./k. Wasser, Kaffeemaschine und Staubsauger, EI-Herd, Kühlschrank und Dunstabzug, Mikrowelle, Badewanne, WC. Warmes und kaltes Wasser, 1-TV, Parabol, Waschmaschine, Gartenmöbel, Doppelbetten: 1, Einzelbetten: 2, Parken auf der Strasse, Gefrierfach. (20 l), Gemeinsamer Aussenpool (105m²)

Beschreibung anzeigen in [deutsch](#) [deutsch](#) [englisch](#) [englisch](#) [ungarisch](#) [ungarisch](#) [rumänisch](#) [rumänisch](#)

Direkt am schönen Gemeinschaftspool liegt ebenerdig dieses komfortable Appartement, von dem man bequem Geschäfte und Restaurants zu Fuß erreichen kann.

Ausstattung

Badezimmer: WC. Warmes und kaltes Wasser. Badewanne.

Badezimmer 2: WC. Warmes und kaltes Wasser. Dusche.

Schlafzimmer: 2 Schlafplätze (Doppelbett)

Schlafzimmer 2: 2 Schlafplätze (Einzelbett)
Terrasse: Balkon und Terrasse

Weitere Räume: Wohnraum/Ki

Baujahr: 2002

+++ Reiserück

+++ Reiselektionskostenversicherung inklusive +++
Beschreibung erscheid in [Tabelle 1](#) deutsch [Tabelle 2](#) englisch

Beschreibung anzeigen in [deutsch deutsch](#) [englisch englisch](#)

Okt 2008							Nov 2008							Dez 2008							Jan 2009							Feb 2009						
Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So
			1	2	3	4				1	2			1	2	3	4	5	6	7	1	2	3	4										
6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14	5	6	7	8	9	10	11	2	3	4	5	6	7	8
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21	12	13	14	15	16	17	18	9	10	11	12	13	14	15
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28	19	20	21	22	23	24	25	16	17	18	19	20	21	22
27	28	29	30	31			24	25	26	27	28	29	30	29	30	31					26	27	28	29	30	31		23	24	25	26	27	28	

Abbildung 4.13.: Detailansicht, tourist-online.de²⁸

termin zu wählen, oder sich die Belegungsübersicht und den Preisspiegel anzusehen. Desweiteren sind Verweise platziert um auf Informationen wie z.B. Ortsbeschreibung, Karten- und Wetteransicht oder Routenplaner zuzugreifen.

Ferienhaus.com listet alle Informationen auf einer Seite auf. Das Informationsangebot ist sehr umfangreich. Die Detailansicht beginnt mit einer Bildergalerie. Durch Überstreichen der Vorschaubilder erfolgt eine Anzeige des gewählten Bildes als Großansicht. Ein Klick auf die Bilder startet zusätzlich eine Diashow in einem Layer. Der nächste Abschnitt wird durch eine Kurzbeschreibung eingeleitet. Um so ausführlicher ist der anschließende Beschreibungstext. Nebenkosten und Entfernung zu wichtigen Orten sind extra aufgeführt und gekennzeichnet. Ihnen folgt eine Auflistung aller Merkmale, die das Objekt zu bieten hat. Eine Spalte auf der rechten Seite nennt Eckdaten zum Haus. Es handelt sich dabei um Hausgröße, max. Belegung, Anzahl spezieller Räumlichkeiten, Mindestpreis, Haustiermitnahme, Poolstatus und Strandentfernung. Ein Feature,

³⁰ <http://www.ferienhaus.com> - Suche - Übersicht - Detailansicht



Meine Auswahlkriterien

Deutschland , Bayerischer Wald,
Bischofsmais , Kamin
Hauscode: BIM106
Belegung: bis 7 Personen
Preis pro Woche ab: € 384

[Auswahlkriterien ändern](#)
[zurück zur Trefferübersicht](#)

Infos

[Ortsinfos](#)
[Mietwagen](#)
[Karte / Wetter](#)
[Routenplaner](#)
[Währungsrechner](#)

Bilder



[weitere Bilder](#)

BIM 102 – Ferienhaus 6 Personen
BIM 103 – Ferienhaus 7 Personen
BIM 105 – Ferienhaus 8 Personen
BIM 106 – Ferienhaus 6-7 Personen

Die schönen Ferienhäuser, Doppelhäuser, stehen ruhig und sonnig im Ortsteil Habischried, am Rande eines kleinen Feriendorfes. Gasthöfe ca. 200 m. Ortsmitte Bischofsmais mit Geschäften 4 km. Parkplätze an bzw. nahe den Häusern.

Die von uns angebotenen Haushälften stehen jeweils auf einem nicht umzäunten Gartengrundstück mit Rasenfläche/Liegewiese und einer sonnigen Terrasse mit Holzbänken, Tisch und Sonnenschirm. Sie sind großzügig geschnitten und gut bis komfortabel eingerichtet mit Kachelöfen bzw. Kamin (€ 6 pro ...)

Anreiseterrasse / Klassik

Anreiseterrasse

Aufenthaltsdauer

Personen

Preis / Belegung
[Belegungsübersicht](#)
[Mietpreise](#)

Mietpreise pro Wohnung und Woche in EUR

innerhalb der angegebenen Saisonzeiten.
Nebenkosten siehe Objektbeschreibung.
Buchungen von **Samstag bis Samstag**.

BIM	106
Personen	7
19.04.2008 - 21.06.2008	384
21.06.2008 - 12.07.2008	514
12.07.2008 - 23.08.2008	644
23.08.2008 - 30.08.2008	514
30.08.2008 - 01.11.2008	384

[Währungsrechner](#)

Belegungsübersicht zu BIM106

Wählen Sie hier Ihren Anreiseterrasse aus.
Mögliche Anreisetage sind unterstrichen.

☐ frei ☐ belegt ☐ nicht buchbar

(Die Übersicht wird alle 30 min aktualisiert, daher Zwischenverkauf vorbehalten.)

Oktober 2008

Mo	Di	Mi	Do	Fr	Sa	So

Abbildung 4.14.: Auszug aus Detailansicht, interchalet.com²⁹

welches bei den bereits genannten Anbietern nicht zu finden war, ist die Kommentierungsmöglichkeit. Urlauber können an dieser Stelle Kommentare zum Objekt hinterlassen. Es besteht für einen Interessenten die Möglichkeit, sich anhand der Kommentare ein Bild vom Objekt und seiner Umgebung zu machen. Der Kommentarbereich ist gefolgt von einer Karte auf der eine geografische Einordnung des Objektes möglich ist. Die Kartenfunktionalität wird wie bei schon erwähnten Anbietern von Google bereitgestellt.

Als weiterer Pluspunkt kann die Möglichkeit gesehen werden einen Films abzurufen. Über den eingebetteten Flash Player³¹ besteht die Möglichkeit, ein Video über das Ferienobjekt anzusehen. Im Film sieht der User sowohl Außenansichten als auch einzelne Räumlichkeiten im Objekt. Es ergibt sich u.U. ein besserer Eindruck als über die Bilder-

³¹ <http://www.adobe.com>

Avignon Umgebung Lorial du Comtat Ferienhaus PRV087

Klicken Sie bitte auf ein großes Bild, um noch mehr und größere Bilder anzusehen

Buchung

Verfügbarkeit

Video

Zur Merkliste

« Vorheriges Objekt

« Zurück

Nächstes Objekt »

Beschreibung PRV087

Dieses Mas bei Lorial ist ein ursprüngliches provenzalisches Landhaus aus dem Jahre 1883 und wurde 2005 komplett renoviert, ohne seinen Charme zu verlieren. Nun finden Sie ein äußerst komfortables Ferienhaus vor mit großen Schlafräumen, neuen Bädern und provenzalischem Flair.

Das Grundstück erstreckt sich über ca. 5000 qm in schattiger und ländlicher Umgebung. Der Mont Ventoux scheint zum Greifen nah, die bekannten Weinorte wie Gigondas oder Vacqueyras sind ein Katzensprung entfernt. Auch Lorial als kleiner Ort hat eine sehr gute Infrastruktur. Alles beste Voraussetzungen für einen erholsamen und eindrucksvollen Urlaub. Das Haus verfügt über eine Wohnfläche von ca. 250 qm, verwindet über 2 Etagen. Im Erdgeschoss befindet sich die gut eingerichtete Küche mit u.a. Dunstabzugshaube, Tiefkühlsch, Geschirrspüler.

Separates WC, Babybett und Hochstuhl. Teilweise Fußboden - und Zentralheizung. Haustiere sind nicht erwünscht. Sitzplatz mit Grill im Garten unter der Platane. Der private, ca. 6 m x 12 m große, ovale Swimmingpool (Sicherheit durch Alarmanlage) liegt etwas höher, es stehen Liegestühle für die breite, sonnige Terrasse zur Verfügung. Ping-Pong-Tisch. Parkplätze auf dem Grundstück.

Entfernungen
zum Kaufmann und Restaurant in Lorial ca. 1 km; Supermarkt 5 km; Tennisplatz ca. 7 km; Golfplatz ca. 20 km; Carpentras ca. 9 km; Orange ca. 18 km; Avignon ca. 25 km; Aix-en-Provence ca. 97 km.

Nebenkosten
Heizung bei Bedarf 120 €/Woche; Endreinigung 120 €; Kautions 500 €. Bettwäsche (8 € p.P./N.) und Handtücher (6 € p.P./N.) können angemietet werden.

☒ Babybett
☒ eingezäuntes Grundstück
☒ Kamin

☒ CD Player
☒ Geschirrspüler
☒ Parkplatz am Haus

**Frankreich - Provence->Avignon
Umgebung->Lorial du Comtat**

Ferienhaus	250 m ²
Personen	10
Schlafzimmer	4

Swimmingpool Ja 6 x 12 m

Strandentfernung (m)

Der nächst größerer Ort Carpentras ist als Kleinstadt der Nabel eines bäuerlichen Umlands. Viele prächtige Bauten stehen noch aus der Papstzeit innerhalb der mittelalterlichen Mauern. Der Höhepunkt der Region bleibt der Mont Ventoux, der 'Windumtoste'. Man erkennt ihn schon aus der Ferne an seinem kahlen Gipfel, er ist mit seinen 1909 m die höchste Erhebung in der Provence. Man kann seine einmalige Pflanzenwelt zu Fuß erkunden oder auf den Spuren der Tour de France den Gipfel mit dem Fahrrad erobern. Dabei werden Sie nicht alleine sein!

Kommentare und Empfehlungen

Marion Kutschank - 30.Okt.2008

Eindrücke unserer Kunden: Unser Kind - und wir auch - fand das Haus klasse, weil wir endlich mit Geschwistern und Cousins ein großes Haus gefunden haben, in dem jeder auch einmal für sich sein konnte.

Kommentare schreiben


Map Satellite Hybrid

Abbildung 4.15.: Detailansicht Auszug 1, ferienhaus.com³⁰

galerie. Eine Tabelle mit Saisonpreisen gibt Auskunft über Kosten zum gewünschten Reisetermin. Ferienhaus.com bietet an dieser Stelle ebenfalls Komfort. Preisangaben können wahlweise in Euro, Pfund, Schweizer Franken oder US Dollar erfolgen. Eine Auswahl ist über das Pulldown Menü möglich.

Neben Kontaktinformationen ist eine Belegungsübersicht Standard. Mit Ausblick auf ein volles Jahr sieht der User sofort in welchen Zeiträumen das Objekt frei oder belegt ist. Eine direkte Buchung auf den Webseiten ist nicht möglich. Anstelle eines Buchungsfomulares sind Kontaktdaten hinterlegt.

³² <http://www.ferienhaus.com> - Suche - Übersicht - Detailansicht



[an Freund senden/a>](#)
[drucken](#)
[Zur Merkliste](#)

Preise:

Mietpreise zzgl. angegebene Nebenkosten EUR Euro

Start	Ende	Tag	Woche	Monat	Beschreibung
15.Jan.2009	23.Mai.2009	--	1.543 EUR	--	
23.Mai.2009	04.Jul.2009	--	1.840 EUR	--	
04.Jul.2009	29.Aug.2009	--	2.977 EUR	--	
29.Aug.2009	12.Sep.2009	--	1.840 EUR	--	
12.Sep.2009	12.Dez.2009	--	1.543 EUR	--	

Verfügbarkeit - PRV087

Verfügbarkeit wird Montag bis Freitag aktualisiert, ist in der Regel also immer aktuell.
 2 Wochen-Lücken in der Plan tafel können auch nur für 2 Wochen gebucht werden.
 In der Nebensaison sind auch 1 Wochen-Buchungen möglich.

■ = Nacht belegt
 Kalender zuletzt aktualisiert: **Mo 03. Nov. 2008**
[Schließen x](#)

Nov 2008							Dez 2008							Jan 2009						
Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr
+	+	+	4	5	6	7	1	2	3	4	5								1	2
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	9
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23
29	30						27	28	29	30	31			24	25	26	27	28	29	30
														31						

Feb 2009							Mär 2009							Apr 2009						
Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr
1	2	3	4	5	6		1	2	3	4	5	6						1	2	3

Abbildung 4.16.: Detailansicht Auszug 2, ferienhaus.com³²

5. Verwendete Systeme und Software

5.1. Entwicklungsumgebung

Bei der IBE handelt es sich um eine Web-Anwendung. Dementsprechend ist es notwendig, die Applikation auf einem Webserver zu entwickeln. Die Anwendung arbeitet mit großen Datenmengen die in Datenbanken erfasst und ausgewertet werden. Der genutzte Webserver muss deshalb in der Lage sein mit Datenbanken zu arbeiten. Da es sich teilweise um enorme Datenmengen handelt, die oft zu aktualisieren sind, wird die Nutzung einer relationalen Datenbank³³ vorgezogen. Nahezu alle Datensätze der Anbieter sind einfach und klar strukturiert. Die Anlieferung erfolgt z.T. schon fertig in einem relationalen Aufbau. Sie besitzt zugleich eine höhere Performance als Objektdatenbanken³⁴, insbesondere was Schreiboperationen betrifft.³⁵ Als Standardlösung bietet sich eine SQL Datenbank an. Im vorliegenden Fall handelt es sich um eine MySQL³⁶ Datenbank, sie stellt eine populäre Variante der SQL Datenbank dar, was sicher nicht zuletzt daran liegt, dass sie „Open Source“³⁷ vertrieben wird. Sie kommt auf den Servern der Firma oft zum Einsatz. Die Web-Anwendung muss auf Eingaben eines Users reagieren, Inhalte dynamisch darstellen und die Möglichkeit besitzen auf die Datenbank zuzugreifen. Auf der anderen Seite, dem sogenannten *Backend*³⁸, muss die Möglichkeit gegeben sein, unterschiedliche Datenformate zu verarbeiten, Datenbanken anzusprechen und mit Schnittstellen der Anbieter zu kommunizieren. Solch eine Aufgabe ist mit HTML³⁹ nicht zu bewältigen. Ein mächtiges Universalwerkzeug stellt hierfür die Scriptsprache PHP⁴⁰ zur Verfügung. Sie beherrscht die Kommunikation mit SQL Datenbanken und kann XML Dateien mit Hilfe der PHP Erweiterung *SimpleXML* verarbeiten. Ihr ist ebenso ein Umgang mit dem SOAP⁴¹ Protokoll möglich.

Ein weiteres Feld stellt die bequeme, unkomplizierte und ressourcenschonende Benutzung der Webseite dar. In den letzten Jahren kam es zur Nutzung eines Systems namens AJAX⁴². AJAX bedeutet soviel wie „Asynchronous Javascript and XML“. Es bietet

³³ Riccardi 2001, 114

³⁴ Riccardi 2001, 580ff

³⁵ Wikipedia Objektdatenbank, Stand: 15. April 2009, aufgerufen am 23. April 2009

³⁶ Riccardi 2001, 205f

³⁷ opensource.org, Stand: 13. März 2007, aufgerufen am 11. Juni 2009

³⁸ Bezeichnet Programmteile deren Prozesse im Hintergrund ablaufen und der Nutzer nicht zu sehen bekommt

³⁹ W3C 2009, <http://www.w3.org/html>

⁴⁰ Kannengiesser 2004, 5

⁴¹ W3C 2007, <http://www.w3.org/2000/xml/Group/>

⁴² Holzner 2006, 27ff

die Möglichkeit Teilbereiche einer Webseite nachzuladen, anstatt die Seite komplett neu aufzubauen.

5.2. Angewendete Systeme

5.2.1. AJAX

AJAX steht für „Asynchronous JavaScript and XML“ und bezeichnet ein Verfahren, welches die asynchrone Datenübertragung zwischen Server und Browser beschreibt. Es bietet den Vorteil, dass innerhalb einer HTML Seite HTTP Anfragen⁴³ durchführbar sind, ohne die Seite komplett neu zu laden. Auf diese Weise lassen sich einzelne Elemente der Seite aktualisieren. Für das Ausführen von AJAX Anweisungen ist eine Interpretation von JavaScript⁴⁴ seitens des Browsers zwingend notwendig. Spezielle Browser Plugins zur Darstellung sind nicht nötig.

5.2.2. PHP

PHP stand ursprünglich für „Personal Home Page Tools“ und wurde später in „Hypertext Preprocessor“ umbenannt.⁴⁵ Die Scriptsprache orientiert sich an der Syntax von C. C wiederum ist eine Programmiersprache für Betriebssystem- und Programmentwicklung⁴⁶. Das Einsatzgebiet von PHP besteht in der Entwicklung von dynamischen Webseiten. Wichtige Merkmale sind die umfassende Unterstützung von Datenbanken, eine Einbindung von Internet Protokollen und die Möglichkeit zusätzliche Bibliotheken einzubinden. Eine Bibliothek ist z.B. das GD-Library oder ImageMagic mit deren Funktionen etliche Grafikoperationen möglich sind. Zu den Standardfunktionen von PHP gehören u.a. folgende Möglichkeiten.:

Arrayfunktionen erstellen, bearbeiten, sortieren und auslesen von Arrays

Stringfunktionen Zeichenketten erstellen, konvertieren, verbinden, trimmen, bearbeiten

Datenbankanbindung Kommunikation mit Datenbanken und der dadurch gegebenen Möglichkeiten Tabellen, anzulegen, auszulesen, editieren oder löschen zu können

Zipfunktionen entpacken und packen von Dateien

FTP-Funktionen Authentifizierung, Up- und Downloads auf bzw. von FTP Servern

⁴³ Jamsa 1996, 95ff

⁴⁴ Lubkowitz 2005, 373f

⁴⁵ Kannengiesser 2004, 16ff

⁴⁶ Wikipedia C (Programmierprache), Stand: 16. Juni 2009, aufgerufen am 18. Juni 2009

5.2.3. SOAP

SOAP war ursprünglich die Bezeichnung für „Simple Object Access Protocol“ und beschreibt im eigentlichen Sinne ein Netzwerkprotokoll.⁴⁷ Es ermöglicht den Datenaustausch zwischen verschiedenen Systemen oder kann Vorgänge auf entfernten Rechnern starten, was als sog. Remote Procedure Call bezeichnet wird. SOAP ist kein eigenständiges System. Vielmehr benutzt es bereits vorhandene Dienste. Zu übertragende Daten werden als XML abgefasst und über Internetprotokolle, die wiederum auf eine Transportschicht wie z.B. TCP⁴⁸ zugreifen, versendet. Seit der Version 1.2 stellt der Begriff SOAP kein Akronym mehr dar, da alle bis dahin verwendeten Deutungen den Umfang von SOAP nicht widerspiegeln.

Das Protokoll wurde nicht von Grund auf neu entwickelt, sondern leitet sich aus XML-RPC ab, welches von Dave Winer entwickelt wurde.⁴⁹ Initiator für SOAP war in diesem Fall die Firma Microsoft. Um daraus einen Standard in der Internetwelt zu schaffen, mussten einige große Computerfirmen davon überzeugt und zur Mitarbeit bewegt werden. Der Plan ging auf und Microsoft reichte SOAP im Jahr 2000 beim W3C ein. SOAP ist als W3C Empfehlung⁵⁰ eingestuft. Wichtiger Bestandteil sind Regeln über den Nachrichtenaufbau und die Interpretation von Nachrichten. Es enthält ebenfalls ein Regelwerk, welches den Aufruf entfernter Prozeduren spezifiziert. Die zu übertragenden Informationen sind dagegen nicht reglementiert.

Im vorliegenden Fall soll SOAP dazu dienen, Anfragen an Ferienhaus Anbieter zu verschicken. Je nach Anfrage liefert der Webservice des Anbieters Informationen über Preise oder Verfügbarkeit eines betreffenden Objektes zurück.

5.2.4. XML

XML bezeichnet die „Extensible Markup Language“ was zu deutsch soviel bedeutet wie „erweiterbare Auszeichnungssprache“.⁵¹ Sie stellt hierarchisch Strukturierte Daten in Form von Text dar. Auf binäre Teile wird bewusst verzichtet, so bleibt sie auch für Menschen lesbar. Ein wichtiger Gedanke bei der Entwicklung von XML war die Trennung von Daten und ihrer Darstellung. Jedes Dokument besteht aus Elementen, die aus einem Start-Tag, der gewünschten Information und einem End-Tag bestehen. Start-Tags können zusätzlich mit Attributen versehen werden. Der logische Aufbau des Dokuments entspricht einer Baumstruktur.

5.2.5. SQL

SQL „Structured Query Language“ bezeichnet eine Datenbanksprache, mit der es möglich ist, Datensätze in relationalen Datenbanken abzufragen, zu manipulieren oder zu erzeu-

⁴⁷ Hauser 2004, 39

⁴⁸ Jamsa 1996, 61ff

⁴⁹ vgl. Hauser 2004, 39f

⁵⁰ W3C 2007, <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>

⁵¹ Lubkowitz 2005, 865ff

gen.⁵² Der Aufbau der SQL Syntax ist einfach gehalten und orientiert sich an englischer Umgangssprache. Eine simple Abfrage aller Daten einer Tabelle sieht deshalb folgendermaßen aus: *SELECT * FROM tabelle* . „Tabelle“ steht hierbei für den Namen der entsprechenden Relation. Das Wildcard steht für die Namen der Spalten in der Tabelle und führt im gezeigten Beispiel zur Ausgabe aller Spalten der Tabelle.

Erzeugung, Manipulation und Löschung sind wichtige Operationen beim Umgang mit Datenbanken. Für einen neuen Eintrag in eine Tabelle existiert der Insert Befehl, *INSERT INTO tabelle...* . Eine Manipulation findet durch den Update Befehl statt, *UPDATE tabelle SET spalte...* . Eine Löschung ist über Delete möglich *DELETE FROM tabelle...* .

SQL stellt in heutiger Zeit einen Quasi Standard dar. Es bildet die Schnittstelle zwischen Anwendung und Datenbank. Aufgrund seiner Position ist in vielen heutzutage verbreiteten Datenbanken eine Implementierung von Teilen des SQL Sprachstandards vorhanden. Anwendungen können auf diese Art und Weise unabhängig vom vorhandenen Datenbanksystem arbeiten. Neben den Standardoperationen besteht die Stärke in der Verknüpfung und des Vergleichs von Tabellen, sog. Relationen, untereinander.

5.2.6. HTML

HTML gilt als Synonym für die „Hypertext Markup Language“ und stellt eine Scriptsprache dar, die zur Erstellung von Webseiten dient.⁵³ Sie ermöglicht die Formatierung von Informationen, wie Bildern und Texten. Des weiteren bietet sie die Möglichkeit Formulare zu erstellen und Objekte einzubetten. Zugleich lässt sich die Zeichenkodierung des Dokumentes definieren, so dass dem Browser bekannt ist in welchem Format Texte darzustellen sind. Das Projekt nutzt z.B. UTF-8 als Codierung. Weiterhin legt es auch den Typ fest. So gibt es Varianten wie *Strict* und *Transitional*. Sie definieren ob bestimmte, in früheren HTML Versionen genutzte, Elemente im Dokument auftauchen dürfen und an welchen Stellen dies erlaubt sind.

⁵² Riccardi 2001, 205f

⁵³ Lubkowitz 2005, 125ff

6. Datenbank

6.1. Einleitung

Zentraler Bestandteil des Systems ist die Datenbank. In ihr laufen alle Anbieterdaten zusammen. Sie bildet zugleich das Rückgrat der Buchungsmaske. Diese bezieht u.a. sämtliche Länder, Regionen und Orte für die Auswahlmenüs und liefert eine ebensolche Auswahl bei direkter Suche. Alle Daten von Objekten, ausgenommen der Bilder, gehen aus ihr hervor und liefern das nötige Material für Objekt- und Belegungsübersicht sowie Detailansicht. Während der Buchung bezieht sich die Maske daraus Angaben über Zusatzleistungen, Währung und Zahlungsmodalitäten. Informationen über den Buchungsvorgang werden in ihr gespeichert.

Für die korrekte Einordnung neuer Datensätze stehen eine Reihe von Referenztabellen bereit. Sie liefern Informationen, um aus den verschiedenen Datenformaten der Anbieter ein einheitliches Format erzeugen zu können. Betroffen sind davon sämtliche Stammdaten, da sie Angaben über die Lage eines Objektes und die zugehörige Objektreferenz besitzen. An die Objektreferenz sind weitere bereits erwähnte Daten geknüpft.

Die Datenbank liefert, wie bereits erwähnt, ebenfalls Datensätze für Auswahlmenüs in der Weboberfläche und Wechselkurse für die Umrechnung einer Landeswährung in Euro. Jeder Tabelle ist ein Bezeichner vorangestellt. Dieser gibt Auskunft für welchen Aufgabenbereich sie verwendet wird.

Um dem System Zukunftssicherheit zu geben, erfolgt die gesamte Zeichencodierung in der Datenbank im Format UTF-8 Unicode⁵⁴.

6.2. Tabellentypen

Menu „Menu“ steht für Tabellen, die genutzt werden um Auswahlmenüs zu füllen. Sie bauen auf den ähnlich lautenden Referenztabellen auf. Der Unterschied besteht darin, dass Referenztabellen unterschiedliche Kennungen mit selber ID besitzen können. Es ist ein notwendiger Umstand um bei Vergleichsoperationen alle möglichen Varianten eines Gebietes zur Hand zu haben. In den Menüs würde dies zu ähnlich lautenden Einträgen führen, hinter denen letztendlich die selbe Referenzierung steckt. Von Zeit zu Zeit ist ein Abgleich der Tabellen notwendig.

Buc Bei „buc“ handelt es sich um Tabellen die mit dem Buchungsvorgang zu tun haben. In ihnen werden z.B. Daten erfasst, die Informationen über Buchungsstatus, das

⁵⁴ Unicode Inc. 2003, 1351

gebuchte Objekt und die buchende Person enthalten. Sie sind für Rückfragen und Klärung von Unstimmigkeiten wichtig.

Obj Tabellen mit „obj“ enthalten immer Daten zu entsprechenden Objekten. In ihnen sind z.B. Merkmale, Zusatzleistungen, Texte, Preise oder Bildverweise zu den Ferienobjekten zu finden.

Ref „Ref“ beschreibt Referenztabelle. In ihnen sind immer Referenzen zu bestimmten Bereichen vorhanden. Referenziert sind z.B. Länder, Orte, Anbieterdaten zu Ferienhaus System oder Codierungen.

Mon Das „mon“ bezieht sich auf Tabellen, die mit Finanzen zu tun haben. Dabei geht es um Dinge wie Wechselkurse oder Kreditkartentypen.

6.3. Beschreibung der Tabellen

6.3.1. buc_personen

Bucht der User ein Ferienobjekt, wird abschließend der Buchungsstatus ausgewertet und ein Eintrag in der Tabelle „buc_personen“ gespeichert. Die Daten liegen temporär vor und beinhalten die Anschrift, E-Mail sowie Telefonnummer als auch den Code des gebuchten Objektes und die Anbieterkennung. Wichtig sind in diesem Zusammenhang die Einträge über Buchungsstatus und Zeitpunkt der durchgeführten Buchung. Sollten später Rückfragen auftreten, sind die Daten ein wichtiger Hinweis darauf, wann das Objekt gebucht wurde und ob die Buchung laut System erfolgreich war. So können bei eventuellen Rückfragen Unklarheiten beseitigt werden.

name	vorname	str	...	objekt	kategorie	datum	zeit	erfolg
Mustermann	Max	Musterstr 132b	...	ES9526.100	1	2008-08-18	11:16:18	N
Mustermann	Max	Musterstr 53	...	BUN04001	F	2008-09-11	10:30:44	Y

Tabelle 6.1.: Buchungserfassung

6.3.2. menu_land, -region, -ort

In diesen Tabellen Liegen jeweils Bezeichnungen der Länder, Regionen und Orte vor. Ihre Angabe wird durch entsprechende ID Einträge ergänzt und eine eindeutige Zuordnung garantiert. Länder besitzen eine zwei- bzw. dreistellige Kennung. Sie ist auf die Abkürzung des Landesnamen zurückzuführen. Für die bessere interne Handhabung bei Ergänzungsoperationen existiert eine zusätzliche Nummerierung.

Regionen sind mit einem vierstelligen Code versehen, der sowohl Region als auch Land repräsentiert. Daneben erhält jeder Eintrag eine Landeskennung und ebenfalls eine interne Nummer.

id	IDRegion	LandID	Region
6	1005	AT	Tirol
23	1205	CH	Wallis
199	2340	IT	Sizilien
207	2505	PL	Tatras
341	3409	SE	Nord-Schweden

Tabelle 6.2.: Menüeinträge für Regionen

Für Orte ist ein siebenstelliger Code vorgesehen. Er liefert neben einer eindeutigen Kennzeichnung des Ortes Informationen über die zugehörige Region und das Land. Nichtsdestotrotz finden sich auch hier weitere Einträge mit Regionscode und Landeskenntung. Die interne ID wird ebenfalls vergeben.

Die Menütabelle leiten sich von den Referenztabellen für Länder Regionen und Orte ab. Im Gegensatz zu den Referenztabellen enthalten sie keine Synonyme, welche dort für Vergleichsoperationen wichtig sind.

6.3.3. mon_creditcard

Bei den meisten Anbietern ist eine Zahlung mit Kreditkarte möglich. Viele Anbieter kürzen die Kreditkartennamen ab, um die spätere Übertragung der Daten zu vereinfachen. Da sowohl Namen als auch Abkürzungen häufiger gebraucht werden erfolgt ihre Angabe nicht in den Scripten. Hier bietet sich die flexible Lösung über die Datenbank an. Neben Namen und Abkürzung besitzt jeder Datensatz eine interne ID.

id	typ	kurz
1	MasterCard	MAST
2	VisaCard	VISA
3	American Express	AX
4	MasterCard	MC
5	VisaCard	VI

Tabelle 6.3.: Tabelle für Kreditkartentypen

Bei einem späteren Ausbau des Systems ist u.U. ein weiterer Eintrag für die Anbieterkennung nötig.

6.3.4. mon_wechselkurs

Einige Länder geben als Währung ihre Landeswährung an und verzichten auf die Angabe von Preisen in Euro. Um einem User die Umrechnung abzunehmen, stellt diese Tabelle Wechselkurse bereit.

Sie beinhaltet sowohl den Namen der ursprünglichen Währung als auch der Währung in die umgerechnet wird. Für die Ursprungswährung gibt es eine zusätzliche Spalte, in der

ein Referenzbetrag eingetragen ist. An diesem orientiert sich der Wechselkurs. Für das Beispiel Schweizer Franken ergab sich im Sommer 2008 für einen CHF ein Wechselkurs von 0,6157. Dies entspricht nach kaufmännischem Runden einem Wert von 0,62 Euro. Zwar wird nach der Berechnung auch ein Betrag für die neue Währung erzeugt, dieser ist jedoch nicht relevant. Wichtig sind immer Bezugswert der ursprünglichen Währung und der Wechselkurs.

waehr_alt	wert_alt	waehr_neu	wert_neu	umrechnung
CHF	1.00	EUR	0.62	0.6157
PLN	1.00	EUR	0.31	0.3069
GBP	1.00	EUR	1.27	1.2662
USD	1.00	EUR	0.68	0.6759

Tabelle 6.4.: Wechselkurse der benötigten Währungen

6.3.5. obj_buhinweise

Für viele Objekte gibt es Buchungshinweise. Sie müssen z.T. beim Ausfüllen des Buchungsformulars extra bestätigt werden um eine Buchung durchführen zu können. Jedes Objekt kann mehrere Hinweise enthalten. Da ein Hinweis manchmal nur zeitlich begrenzt gültig ist gibt es die Möglichkeit zu jedem Datensatz ein Anfangs- und Enddatum festzulegen. Einträge die zeitlich unbegrenzt sind enthalten als Datumsangaben „0000-00-00“

IDAnbieter	ObjReferenz	Kategorie	Buhinweis	Datumvon	Datumbis
1	IT9101.100	1	Pool nicht verf...	2008-07-28	2008-09-27
1	ES9456.637	1	Baustelle auf ...	2007-11-30	2008-09-30
2	COC01074	0	Eine Kautiion ...	0000-00-00	0000-00-00
2	SSD05009	0	Eine Kautiion ...	0000-00-00	0000-00-00

Tabelle 6.5.: Buchungshinweise für einzelne Objekte

6.3.6. obj_hauspic, obj_wohnungspic

In diesen Tabellen sind zu jedem Objekt Bilder verlinkt, die der Anbieter bereit stellt. Für Hausbilder ist nur der Anbieter- und Referenzcode des Objektes nötig. Wohnungsbilder erfordern zusätzlich eine Angabe der Wohnungsnummer. Jedes Objekt und jede Wohneinheit besitzen mehrere Bilder. Um eine spätere Verarbeitung zu erleichtern, erfolgt während der Aufbereitung der Daten für die Datenbank eine Nummerierung aller Bilder pro Objekt bzw. Wohneinheit. Den Bildern kann bei Bedarf eine Beschreibung hinzugefügt werden, dazu steht eine Spalte in der Tabelle bereit. In der Tabelle für Wohnungsbilder ist sogar eine Saisonangabe möglich. Hier wird zwischen „W“ und „S“, also Winter und Sommer unterschieden.

IDAnbieter	ObjReferenz	Bildnummer	Bildtext	Bildlink
1	PL8330.100	4	PL8330.100d.s.jpg	http://...
1	HR4013.121	8	HR4013.121h.s.jpg	http://...
2	ITO06180	10	NULL	http://...
2	BRE03661	4	NULL	http://...

Tabelle 6.6.: Bildverweise für einzelne Objekte

6.3.7. obj_haustext, obj_wohnungstext

Hier befinden sich Beschreibungstexte über das gesamte Haus oder einzelne Wohneinheiten. Um flexibel für Erweiterungen zu sein, steht jedem Beschreibungstext eine Sprachangabe zur Seite. Momentan sind nur Texte mit der Sprachangabe „de“ vorhanden, da die Ausgabe momentan nur in Deutsch erfolgen soll.

IDAnbieter	ObjReferenz	Kategorie	Sprache	Beschreibung
1	FR3420.650	5	de	5-Zimmer- Chalet 95 m2, sehr einfach ...
1	ES9730.966	3	de	4-Zimmer-Wohnung 125 m2 auf 2 Stockwerken ...
1	HU8637.951	1	de	6-Zimmer- Haus 170 m2, komfortabel und schön ...
1	IT2811.100	19	de	3-Zimmer-Wohnung 45 m2, vollständig renoviert ...

Tabelle 6.7.: Beschreibungstexte der Wohnungen

6.3.8. obj_merkmale

Die Tabelle für Merkmale ist eine der umfangreichsten im System. Ziel ist es, eine große Palette an Merkmalen zu erfassen und damit ein Angebot zu ermöglichen, welches bisherige Anbieter am Markt nicht bereit stellen.

Angaben erfolgen zur Inneneinrichtung, Eigenschaften des Hauses, Ausstattung von Hof und Garten, Freizeit- und Erholungsangeboten in der Umgebung, Bewertung vom Haus und maximaler Belegung.

Beispielhaft sind folgende Merkmale für ein Objekt möglich. Es besitzt Geschirrspüler, Klimaanlage und eine Garage. Ein Pool, Sauna und Solarium befinden sich im Haus. Der Bau ist behindertengerecht. Haustiere sind erlaubt. In näherer Umgebung (Angaben von 100m bis 10km sind möglich) befindet sich ein Angelplatz, Badensee oder das Meer. Bewertet ist es mit 3 von 5 Sternen. Die Ausstattung bekommt 3 von 5, die Umgebung 4 von 5 Punkten. Maximal sind 7 Personen erlaubt, wovon 2 Kinder sein dürfen. Kurzaufenthalte werden angeboten.

Die Codierung der Merkmale erfolgt mit den Werten 1 oder 0, wobei 1 für vorhanden und 0 für nicht vorhanden steht. Ausnahmen bilden die Entfernungsangaben und die maximal mögliche Belegung eines Objektes als auch Bewertungskriterien.

IDAnbieter	ObjReferenz	Kategorie	Haustier	Pool	Personen	Zimmer	...
1	FR8419.4	1	1	0	5	3	...
1	DK1091.6	1	0	0	4	2	...
2	TFS01001	ZYYO	0	0	2	0	...
2	IVN01416	CYYO	0	0	5	0	...

Tabelle 6.8.: Merkmale der Objekte

6.3.9. obj_preise

Hier finden sich die Preisangaben der Objekte wieder, dabei sind mehrere Konstellationen möglich. Zum einen entscheidet die Aufenthaltsdauer über den Preis, zum anderen ist er auch vom gewählten Anreisedatum abhängig. Als Aufenthaltsdauer sind meist 1 bis 4 Wochen festgelegt. Eine Anreise ist oft nur am Samstag möglich. Typischerweise ergeben sich so für jede Woche im Jahr 4 Preisangaben. Theoretisch kann sich damit jede Woche der Preis für 1, 2, 3 oder 4 Wochen Aufenthalt ändern. Aufgrund der Möglichkeiten ergibt sich leider ein hohes Datenaufkommen. Da viele der Preisangaben pro Objekt meist über etliche Wochen konstant bleiben, ist die Möglichkeit vorgesehen für Preise eine Zeitspanne anzugeben. Sie beginnt mit dem Datum, ab dem der neue Preis gilt und endet wenn ein weiterer, abweichender Preis auftritt. Selbstverständlich listet eine der Spalten dann den jeweiligen Preis. Die Auswahl findet bei einer Abfrage durch Eingrenzung des Datums statt. Eine Zuordnung zur entsprechenden Wohneinheit erfolgt über Anbieterkennung, Objekt Referenzcode und Wohnungskennung.

IDAnbieter	ObjReferenz	Kategorie	Aufenthalt	Datumvon	Datumbis	Preis
1	DE8222.100	2	28	2009-06-06	2009-07-04	1718.00
1	AD1500.100	2	14	2009-07-25	2009-08-22	2340.00
2	CHB01001	F	14	2008-09-06	0000-00-00	907.00
2	AUV01029	F	7	2008-09-20	0000-00-00	537.00

Tabelle 6.9.: Preise der Objekte

6.3.10. obj_stammdaten

Die Stammdaten bilden den Schlüssel zu vielen Bereichen in der Datenbank. Sie stellen eine Verknüpfung zwischen den Objekten und ihrer geografischen Lage her. Eine geografische Suche und Zuordnung der Objekte ist ohne Stammdaten unmöglich.

Sie enthält eine Kennung des jeweiligen Anbieters, die Referenzierung des Hauses und der zugehörigen Wohneinheiten, verknüpft mit Angaben über Herkunftsland, Region und Ort. Länder, Regionen und Orte liegen wiederum in ihrer Codierung vor und lassen sich über die Referenztabelle für Länder, Regionen und Orte ermitteln.

IDAnbieter	ObjReferenz	Kategorie	Land	Region	Ort
1	CH7500.116	4	CH	1209	1209027
1	AT6313.300	6	AT	1005	1005023
2	IDO01275	CYYYQ	IT	2344	2344015
2	BRE03698	F	FR	1706	1706213

Tabelle 6.10.: Stammdaten

6.3.11. obj_vakanzen

In dieser Tabelle liegen Belegungsdaten der Objekte, ergänzt durch Angaben über Mindestaufenthalt und Schlüsselübergabe, vor. Jeder Datensatz deckt den Zeitraum von einem Monat ab. Beginn ist immer der erste des Monats. Die Angabe erfolgt in der Form „YYYY-MM-DD“ und stellt sicher, dass zwischen Monaten des aktuellen und des Folgejahres unterschieden wird.

Die Belegung selber ist als Zeichenkette codiert. Jedes Zeichen stellt einen Tag im Monat dar, deshalb bilden die Zeichenketten nur Längen von 31, 30 oder 28 bzw. 29 Zeichen. Hinter jedem verwendeten Zeichen steckt eine eindeutige Aussage.

Belegungen sind mit den Zeichen „Y“, „N“ und „Q“ codiert. Das Y signalisiert Verfügbarkeit, ein N zeigt an, dass das Objekt gebucht ist. Bei einem Q besteht nur die Möglichkeit einer Anfrage. Die Übergabe nutzt die Zeichen „X“, „C“, „O“ und „I“. Für mit X gekennzeichnete Bereich ist weder eine Übergabe noch Übernahme möglich. Ein C bedeutet das genaue Gegenteil, Übernahme und Übergabe sind möglich. Das O zeigt eine Übergabe mit Checkout, aber keine Möglichkeit der Übernahme an. Ein I steht dementsprechend für eine Übernahme mit Checkin aber keine Übergabe.

Angaben über den Mindestaufenthalt benötigen eine ganze Reihe an Zeichen. Zur Codierung verwandt werden „0“ und „A“ bis „Z“, wobei 0 für nicht buchbar und A einen Mindestaufenthalt von einem Tag angibt. Jeder folgende Buchstabe im Alphabet stellt eine um eins höhere Anzahl an Tagen des vorherigen Buchstabens dar. Der Buchstabe Z ist z.B. stellvertretend für eine Mindestaufenthaltsdauer von 26 Tagen.

IDAnbieter	ObjReferenz	Kategorie	Datum	Vakanz	Uebergabe	Minstay
1	HR5370.115	1	2009-04-01	NNYYYYY...	XXXICCC...	GGGGGGG...
1	DK1348.35	1	2010-01-01	YNNNNNN...	XOXXXXX...	GGGGGGG...
99	ONFH000065	HA	2009-02-01	YYYYYYY...	CCCCCCC...	CCCCCCC...
99	ONFH000105	WH	2009-02-01	YYYYYYY...	CCCCCCC...	CCCCCCC...

Tabelle 6.11.: Vakanzen der Objekte

6.3.12. obj_zusatzleistungen

Hier sind alle Leistungen erfasst, welche im Preis inbegriffen sind, zusätzlich gebucht werden können oder standardmäßig dazu gehören. Es erfolgt eine Angabe von Leistungsart,

Preis und Währung. Daneben befinden sich Informationen über Serviceart und Berechnungsart in der Tabelle. Die Berechnungsart gibt an, in welcher Art und Weise die Leistung während des Aufenthaltes berechnet wird. Gängige Varianten sind z.B. eine Berechnung pro Woche, pro Tag oder pro Person und Tag. Die Serviceart gibt an ob eine Leistung im Preis inbegriffen oder zusätzlich buchbar ist oder einen Pflichtbestandteil darstellt (z.B. Kautions) und in jedem Fall bezahlt werden muss. Natürlich ist auch hier jede Leistung einem bestimmten Anbieter und Objekt zugeordnet.

IDA...	ObjReferenz	Kategorie	Leistung	Preis	Waehrung	Serviceart	Berechnung
1	PL7809.100	1	Kautions bar	500.00	PLN	Vor Ort zu	pauschal
1	IT5492.810	2	Haustier	50.00	EUR	Zus. buchbar...	pro Woche
2	UFL04023	OYYYO	Babybett	32.00	EUR		Woche/Einheit
2	MLA01002	CYYYP	Frühstück	4.00	EUR		Tag/Einheit

Tabelle 6.12.: Zusatzleistungen einzelner Objekte

6.3.13. ref_interhome

Hierbei handelt es sich um die Referenztabelle für Interhome Objekte. Ihre Aufgabe ist es Daten bereit zu stellen um, eine Konvertierung in das interne Format durchzuführen. Zu jedem Land, jeder Region und jedem Ort ist sowohl die Codierung des Anbieters, als auch die interne Codierung des neuen IBE Systems angegeben. Spalten die mit „ih“ beginnen, stellen Interhome Codierungen dar - solche mit „fh“ die interne Codierung. Die Spalten Land, Region und Ort geben den entsprechenden Namen an. Als Beispiel sei die Region Niederösterreich genannt. Interhome verwendet als Kennung die 10, während ferienhaus.de die 1002 benutzt. Bei einer Aktualisierung, der durch Interhome gelieferten Stammdaten, wird auf diese Tabelle zurückgegriffen.

fh_land	ih_land	Land	fh_region	ih_region	Region	fh_place	ih_place	Ort
AT	AT	Österreich	1002	10	Niederö...	1002001	2042	Weinviertel
CH	CH	Schweiz	1205	10	Wallis	1205022	3901	Saas Grund
DE	DE	Deutschland	1409	9	Eifel	1409001	5100	Aachen
FR	FR	Frankreich	1708	47	Aquitaine	1708010	3353	Hourtin
IT	IT	Italien	2321	44	Toscana	2321009	5253	Pontassieve

Tabelle 6.13.: Referenztabelle für Interhome Objekte

6.3.14. ref_wolters

Die Referenztabelle des Anbieters „TUI-Wolters“ basiert auf dem selben Prinzip wie „ref_interhome“. Im Unterschied dazu beginnen Spalten, die den Anbieter betreffen, mit „wo“ und enthalten Codierungen die Wolters benutzt. Die Bezeichnung „fh“ für die

interne Codierung bleibt gleich. Länder, Regionen und Orte werden ebenfalls mit ihren Namen bezeichnet.

Wolters liefert neben Regionen auch Gebietsdaten. Sie bilden eine Zwischenstufe zwischen Regionen und Orten. Da eine spätere Nutzung nicht ausgeschlossen ist, werden auch diese in der Tabelle erfasst. Zum jetzigen Zeitpunkt besitzen sie jedoch weder eine interne Referenzierung noch sind sie für weitere Vorgänge nötig.

fh_land	wo_land	Land	fh_region	wo_region	Region	...	fh_city	Ort
PT	P	Portugal	2605	ALG	Algarve	...	2605001	Lagos
FR	F	Frankreich	1729	ALP	franz. Alpen	...	1729002	Le Grand...
ES	E	Spanien	1627	CON	Nördliches M.	...	1627003	Salou
GB	GB	Großbritannien	1809	GBE	England	...	1809008	Hayle
IT	I	Italien	2342	IVN	Venetien	...	2342002	Caorle

Tabelle 6.14.: Referenztable für Wolters Objekte

6.3.15. ref_onferienhaus

Hierbei handelt es sich um die Referenztable für die alten ferienhaus.de Ferienobjekte. Die Referenzierung erfolgt in gleicher Weise wie bei Interhome oder Wolters Objekten. Referenzen zum Ferienhaus System sind mit „fh_“ gekennzeichnet. Die vom alten Ferienhaus System mit „on_“. Die „on_“ Felder sind jedoch leer, da die Originalreferenz nicht übernommen wird. Es reichen die jeweiligen Namen. Dennoch kann nicht ausgeschlossen werden, dass sie zu einem späteren Zeitpunkt zum Einsatz kommen.

6.3.16. ref_land, ref_region, ref_ort

fh_land	on...	Land	fh_region	on...	Region	fh_place	...	Ort
ES	0	Spanien	1618	0	Costa Brava	1618016	...	Pals
DE	0	Deutschland	1437	0	Sachsen-Anhalt	1437001	...	Freyburg
FR	0	Frankreich	1737	0	Cote-Azur	1737001	...	Les Issambres ...
HU	0	Ungarn	2109	0	Puszt	2109001	...	Reichelhausen
IT	0	Italien	2345	0	Imperia	2345001	...	Imperia

Tabelle 6.15.: Referenztable für alte Ferienhaus Objekte

Diese drei Referenztabellen sind für das gesamte System eine Notwendigkeit. In ihnen sind die internen Codierungen definiert. Gleich mehrere Anwendungen greifen darauf zurück. Als Beispiele sind die Stammdatenerfassung, Suchfunktion und Menüauswahl zu nennen. Während der Erfassung und Verarbeitung von Stammdaten erfolgt sogar ein automatisiertes Update auf die Tabellen für Regionen und Orte. Sämtliche Referenzen auf Länder, müssen bereits im Vorfeld zur Verfügung stehen, um einen reibungslosen

Ablauf zu gewährleisten. Die Sicherheit erhöht sich, wenn in jeder der Tabellen auch Synonyme mit selber ID erfasst sind.

In der Referenztable der Länder ist sowohl der Name des Landes als auch seine landestypische Abkürzung und eine ID zu finden. Regionen sind in ihrer Referenztable mit Namen, ID für die Region und der ID des Landes gekennzeichnet. Die Referenztable für Orte enthält neben dem Namen des Ortes eine ID für den Ort, die ID der Region und des Landes. Die Spalte „Skiort“ wird nicht mehr benötigt.

IDOrt	RegionID	LandID	Skiort	Ort
1001001	1001	AT	0	Wien
1201006	1201	CH	0	St-Gingolph
1302011	1302	CZ	0	Beroun
1727007	1727	FR	0	Centuri

Tabelle 6.16.: Referenztable für Orte

6.3.17. ref_zusatzleistungen

In der Referenztable für Zusatzleistungen liegen alle bekannten Zusatzleistungen der Anbieter in Textform vor und sind mit einer ID versehen. Spalten deren Bezeichnung mit „c“ beginnt enthalten die Codierung die der jeweilige Anbieter benutzt. Hier ein Beispiel.: Ein Babybett besitzt intern die ID „2“. In der Spalte „c_wolters“ ist der Eintrag mit „A“ codiert.

IDZusatz	Beschreibung	c_wolters
3	Bettwäsche	B
10	Frühstück	D
19	Hochstuhl	H
24	Klimaanlage	I

Tabelle 6.17.: Referenztable für Zusatzleistungen

7. Datenformate der Anbieter

7.1. Format und Struktur der Interhome Datensätze

Interhome liefert seine Daten im CSV⁵⁵ Format aus. Theoretisch ist eine sofortige Übernahme ohne Konvertierung möglich. Das CSV Format stellt eine vereinfachte Tabellenstruktur dar und kann von Programmen wie Microsoft Excel⁵⁶, Open Office⁵⁷ oder einer SQL-Datenbank⁵⁸ verarbeitet werden.

Die Trennung der Zeilen erfolgt durch das Escape⁵⁹ Zeichen für den Zeilenumbruch. Einzelne Spalten sind meist durch eine Kombination aus Anführungszeichen und Semikolon voneinander getrennt. Interhome trennt in seinen Datensätzen nur mit Semikolon.

Der Anbieter unterscheidet Datensätze nach Typ. Jeder Typ liegt in einer separaten Datei vor. Wichtigster Typ sind die Stammdaten. Sie stellen die Basis dar und referenzieren auf alle anderen Datentypen. Sie enthalten zu jedem Objekt eine Referenznummer und die zugehörigen Codierungen für eine Einordnung nach Land, Region und Ort. Zusätzlich listet Interhome in den Stammdaten alle Merkmale des betreffenden Objektes auf. Da es eine interne Codierung für die Lagedaten gibt stellt der Anbieter drei Dateien bereit, die jeweils die Länder, Regionen und Orte aufschlüsseln.

Für jedes Objekt liegt eine Hausbeschreibung - für jede Wohnung im Objekt eine Wohnungsbeschreibung vor. Beide Typen in getrennten Dateien. An erster Stelle steht der Referenzcode des Objektes bzw. der Wohnung. Ohne ihn wäre eine Zuordnung unmöglich. Interhome bietet die Beschreibungen in mehreren Sprachen an und gibt für eine spätere Auswahl in einer zusätzlichen Spalte die Sprachcodierung an. Sie bietet für eine weitere Verarbeitung eine ungemeine Hilfe.

Eine ähnliche Struktur weisen die Dateien für Haus- und Wohnungsbilder, Vakanzen⁶⁰, Preise, Zusatzleistungen und Hinweise auf. Für einige Dateien gibt es zusätzliche Codierungslisten, die Beschreibungen zu den entspr. Codierungen enthalten. Eine genauere Betrachtung findet später im jeweiligen Verarbeitungsfall statt.

⁵⁵ Comma Separated Value. Speichert Tabelleninhalte in Textform und trennt Spalten und Zeilen mit definierbaren Zeichen.

⁵⁶ <http://www.microsoft.com>

⁵⁷ <http://www.openoffice.org>

⁵⁸ Lubkowitz 2005, 723

⁵⁹ Wikipedia Escape-Sequenz, Stand: 27. Januar 2009, aufgerufen am 7. April 2009

⁶⁰ Der Begriff Vakanzen beschreibt den Belegungsstatus des Objektes über einen Zeitraum

7.2. Format und Struktur der Wolters Datensätze

Wolters liefert seine Dateien als XML, der sog. „Extensible Markup Language“, aus. XML ist eine auszeichnende Sprache⁶¹ und als Standard in den W3C Spezifikationen des *World Wide Web Consortium*⁶² definiert. XML kann nicht ohne vorherige Verarbeitung in eine SQL Datenbank geladen werden. Im praktischen Teil der Arbeit findet eine Verarbeitung mit Hilfe von *SimpleXML*⁶³ statt.

Unverzichtbar ist auch hier die Stammdaten-Datei. Innerhalb vom Wurzelement gibt es etliche Knoten mit der Bezeichnung „object“. Sie enthalten alle wichtigen Daten zu einzelnen Ferienobjekten. In ihnen sind u.a. Referenzen in Bezug auf Lage, der Referenzcode und eine Codierung der Kategorie zu finden. Zusätzlich sind Informationen über die maximal zulässige Anzahl an Erwachsenen und Kindern, die untergebracht werden können, hinterlegt. In allen Datensätzen findet sich außerdem der Ortsname und teilweise eine Angabe zum Namen des Objektes. Da auch Wolters die Lagedaten codiert, sind entsprechende Zusatzinformationen nötig. Die jeweiligen Länder, Regionen und Orte finden sich in einer separaten XML Datei wieder.

Im Gegensatz zu Interhome bietet Wolters Beschreibungstexte nur für die Wohneinheiten an. Extra Hausbeschreibungen gibt es nicht. Alle Texte liegen nur in deutscher Sprache vor. Eine eindeutige Zuordnung der Texte erfolgt über den Referenz- und Kategoriecode. Es gibt jedoch Typen für die der Referenzcode ausreichend ist. Als Beispiel sind die XML Dateien für Zusatzleistungen, Zeitrabatte und Bilder zu nennen.

Generell gilt jedoch, dass für eine korrekte Einordnung der Angaben mindestens ein Referenzcode angegeben werden muss. Anders sind z.B. Merkmale, Bildlinks und Zusatzleistungen nicht zuzuordnen.

7.3. Datenformat und Struktur der Privateinträge des ursprünglichen ferienhaus.de Portals

Ferienhaus.de bietet schon seit geraumer Zeit Nutzern einen Eintrag ihres Ferienobjektes auf dem Portal an. Informationen über die Objekte liegen nur in der Datenbank vor, eine Dateistruktur wie bei bereits erwähnten Anbietern gibt es nicht. Es besteht die Notwendigkeit ein Script zu entwerfen welches die Datenbanken abfragt und aus den Informationen die entsprechenden Dateien generiert. Die momentane Informationsverteilung in der Datenbank folgt dem Grundprinzip der bereits genannten Anbieter. Es existieren drei Tabellen für die Lagezuordnung. Je eine für Land, Region und Gebiet. Sie enthalten den Namen der Gebiete und eine ID als Codierung für jeden Eintrag. In den Stammdaten steht dann nur die ID. Der Unterschied zu den beiden genannten Anbietern besteht in der Fülle an Informationen, die bereits in den Stammdaten zusammengetragen sind. Neben einer Vielzahl von Merkmalen enthält sie ebenfalls Links zu Vorschaubildern, Beschreibungen der Wohnung, Zusatzleistungen und eine Möglichkeit

⁶¹ W3C XML 1.0 (Fifth Edition) 26.11.2008

⁶² <http://www.w3.org>

⁶³ Achour 2009, Funktionsreferenz - XML-Manipulation - SimpleXML

zur individuellen Angabe von zusätzlichen Merkmalen. In weiteren Tabellen finden sich Belegungsübersichten, Anfahrsbeschreibungen und Bildergalerien. Eine Referenzierung auf das entsprechende Objekt erfolgt mit Hilfe der ID.

Der Praxisteil beschreibt den Entwurf eines Scriptes, dass die Informationen aus der Datenbank liest und mehrere CSV Dateien erstellt. Jede Datei stellt einen Typ dar und enthält entsprechende Informationen. Da sie bereits nach den Vorgaben der IBE Datenbank generiert werden, erfolgt später ein direktes Einlesen in die neue Datenbank.

8. Datenaggregation

8.1. Vorwort

Eine Datenbank bzw. Datenpool macht nur dann Sinn, wenn in ihm auch Daten gespeichert sind. Daten können entweder per Hand oder automatisiert in die Datenbank gelangen. Eingaben per Hand sind z.B. dann nötig, wenn Daten erst digitalisiert werden müssen. Private Ferienhausangebote auf dem alten ferienhaus.de Portal gelangen auf diese Art und Weise in die Datenbank. Selbstverständlich ist eine manuelle Übertragung von einem auf ein anderes elektronisches System möglich. Aufgrund des Datenvolumens ist sie jedoch unpraktikabel und sprengt den Zeitrahmen. Eine automatisierte Variante ist die bessere Wahl.

Als Hürde stellen sich die verschiedenen Formate und Zusammensetzungen von Daten der Anbieter dar. Je nach Anbieter liegen Datensätze im Format XML oder CSV vor.

Theoretisch besteht die Möglichkeit CSV Dateien direkt in die Datenbank einzulesen. Eine Definition für XML ist bereits im Kapitel 5 Technische Parameter gegeben. Datensätze im XML Format müssen in jedem Fall konvertiert werden.

Anbieter liefern ihre Daten naturgemäß nicht in der Struktur wie sie die Datenbank des neuen Systems aufweist. Deshalb ist es unerheblich in welcher Art und Weise die Daten vorliegen. Eine entsprechende Umstrukturierung und damit Bearbeitung der Daten ist in jedem Fall notwendig. Erschwerend kommt der enorme Umfang der Daten hinzu. Nicht selten umfassen einige gelieferte Daten mehrere hundert Megabyte. Eine Verarbeitung dieser großen Datenmengen führt oft zum Erreichen der Kapazitätsgrenzen. Deshalb mussten mehrere Anläufe unternommen werden um eine Methode zu finden, die mit den Ressourcen sparsam umgeht.

8.2. Bezug der Anbieterdaten

Die Anbieter stellen ihre Daten in gepackter Form auf FTP⁶⁴ Servern zur Verfügung. Nur nach einem Vertragsabschluss mit dem jeweiligen Anbieter ist ein Zugang möglich. PHP bietet Funktionen um Verbindungen zu FTP Servern aufzubauen und einen Datentransfer durchzuführen. Die Funktion *ftp_connect*⁶⁵ wählt den gewünschten Server an. Über *ftp_login*⁶⁶ erfolgt die Anmeldung am Server. Der Login setzt einen Connect voraus. Nach erfolgreicher Authentifizierung ist ein Download über die Funktion *ftp_get*⁶⁷

⁶⁴ Lubkowitz 2005, 58

⁶⁵ Achour 2009, Funktionsreferenz - Sonstige Dienste - FTP - FTP Funktionen - *ftp_connect*

⁶⁶ Achour 2009, Funktionsreferenz - Sonstige Dienste - FTP - FTP Funktionen - *ftp_login*

⁶⁷ Achour 2009, Funktionsreferenz - Sonstige Dienste - FTP - FTP Funktionen - *ftp_get*

möglich. Sie benötigt sowohl den Namen der Quell- als auch eine Bezeichnung der Zieldatei. Der Name der Zieldatei kann eine Verzeichnisangabe enthalten und bietet die Möglichkeit eine Datei vom FTP Server direkt in ein bestimmtes Verzeichnis zu kopieren. Nach erfolgreichem Transfer ist eine Weiterverarbeitung auf dem lokalen Server möglich.

Die Daten liegen nun lokal und in gepackter Form vor. Eine Komprimierung erfolgt häufig im gz oder zip⁶⁸ Format. Für ZIP Dateien stellt PHP die *ZipArchive* Klasse bereit. Nach erfolgreicher Initialisierung ist über den Aufruf *open* aus der Klasse ein Öffnen der Zip Datei möglich.⁶⁹ Im Erfolgsfall entpackt die Funktion *extractTo* alle Dateien des Archivs in das angegebene Verzeichnis. Sind alle Dateien entpackt wird über *close* das Archiv wieder geschlossen. Um nicht unnötig Ressourcen zu verschwenden findet im Anschluss eine Löschung der temporären Zip Datei statt. Zusätzlich ermittelt *filesize*⁷⁰ die Größe der entpackten Datei. Ihre Angabe ist notwendig, um später korrekt mit der *fopen*⁷¹ Funktion arbeiten zu können. Ab diesem Zeitpunkt ist eine Bearbeitung der Daten und eine damit verbundene Anpassung möglich. Sollte ein Öffnen der Datei nicht möglich sein entfällt das Entpacken. An deren Stelle kommt es zur Ausgabe einer Fehlermeldung.

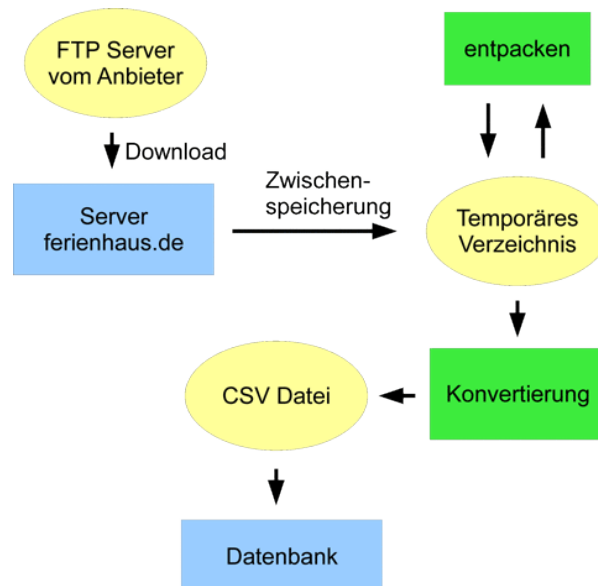


Abbildung 8.1.: Ablauf der Datenaggregation

⁶⁸ Gailly 2003, <http://www.gzip.org>

⁶⁹ Achour 2009, Funktionsreferenz - Erweiterungen zur Datenkompression und Archivierung - ZIP

⁷⁰ Achour 2009, Funktionsreferenz - Dateisystemrelevante Erweiterungen - Dateisystem

⁷¹ Achour 2009, Funktionsreferenz - Dateisystemrelevante Erweiterungen - Dateisystem - *fopen*

8.3. Verarbeitung der Daten

8.3.1. Bearbeitung von Datensätzen im CSV Format

Folgende Idee wurde für die Bearbeitung der CSV Dateien, wie sie beispielsweise Interhome liefert, verwirklicht.

Unter den unzähligen Funktionen von PHP gibt es die Funktion *explode*⁷², um Zeichenketten an bestimmten Stellen zu trennen und ihr Ergebnis in einem Array⁷³ zu erfassen. Jeder Datensatz einer CSV Datei belegt nur eine Zeile. Als Trennung kommt dementsprechend der Zeilenumbruch in Frage. Hier setzt *explode* an. Es erzeugt ein Array und legt jede Zeile in einem Feld ab. Am Ende existiert ein Array mit einer Anzahl an Feldern, die der Anzahl von Zeilen des Ursprungsdokumentes entspricht. Jede Zeile enthält eine endliche Anzahl an Spalten, die wiederum durch ein spezielles Zeichen voneinander getrennt sind. Innerhalb einer While Schleife⁷⁴ folgt ein Aufruf jeder Zeile. Kurz bevor die While Schleife startet wird eine leere Datei geöffnet. Sie enthält am Ende des Schleifenaufufes die bearbeiteten Datensätze. Ein erneuter Aufruf von *explode* trennt nun die Spalten und erzeugt ein weiteres Array welches die Spalteneinträge enthält. Eine Trimmfunktion entfernt aus den im Array befindlichen Einträgen unnötige Leerzeichen und Zeilenumbrüche und verhindert spätere Fehlinterpretationen in der Datenbank. Es folgt eine neue Zusammensetzung der Arrayeinträge im Stil der CSV Formatierung. Die Werte werden in der neuen Reihenfolge, getrennt durch CSV konforme Trennzeichen, und einem Escape Zeichen für den Zeilenumbruch als String in einer Variable⁷⁵ erfasst. Über die Funktion *fwrite*⁷⁶ gelangt der erzeugte String in die Datei. Innerhalb der While Schleife zählt ein Zähler mit. Er ist nötig, um jede Zeile die im Array liegt aufrufen zu können. Sind alle Zeilen verarbeitet, enthält die neue CSV Datei sämtliche Daten und kann direkt in die Datenbank übernommen werden. Alternativ zur While Schleife ist eine Anwendung der Foreach⁷⁷ Schleife möglich. Es entfällt dann das vorher nötige Ermitteln der Zeilenanzahl.

Listing 8.1: Datenkonvertierung Methode 1

```

1 {
2   ...
3   $handle = fopen($objektverz.$dateiname, "r");
4   $contents = fread ($handle, $dateigroesse );
5   $contentneu = strtr($contents, '"', '_');
6   $zeile = explode("\n", $contentneu);
7   $eintraege=count($zeile);
8   fclose ($handle);
9   while ($i < $eintraege) {
10    $spaltenwerte = explode(";", $zeile[$i]);
11    $eintraege_zeile = count($spaltenwerte);
12    $i++;
13    $objreferenz = trim($spaltenwerte[0]);

```

⁷² Achour 2009 Funktionsreferenz - Textverarbeitung - Zeichenketten - explode

⁷³ Achour 2009, Funktionsreferenz - Variablen- und typbezogene Erweiterungen - Arrays

⁷⁴ Holzner 2006, 83

⁷⁵ Ein flüchtiger Speicherort für Informationen, während des Scriptablaufes.

⁷⁶ Achour 2009, Funktionsreferenz - Dateisystemrelevante Erweiterungen - Dateisystem - fwrite

⁷⁷ Achour 2009, Sprachreferenz - Kontrollstrukturen - foreach

```

14  $bildtext = trim($spaltenwerte[1]);
15  $bildlink = trim($spaltenwerte[3]);
16  if ($objreferenz != '' OR !preg_match("/\b$bildtext\b/i", $bildtext_alt)) {
17      $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.
18      $trenner.$bildnummer.$trenner.$bildtext.$trenner.
19      $bildlink.$trenner2."\n";
20      fwrite($handle_writefile, $eintragen);
21  }
22  $objreferenz_alt = $objreferenz;
23  $bildtext_alt = $bildtext;
24  }
25  ...
26  }

```

Eine Verarbeitung nach dieser Methode ist nur mit Quellmaterial möglich, dass nur wenige Megabyte an Informationen umfasst. Größere Datenmengen schöpfen schnell die Kapazitäten aus und benötigen oft mehrere hundert Megabyte an Arbeitsspeicher. Der Nachteil entsteht durch die Art der Verarbeitung. Die Betrachtung der Quelldaten findet immer im Gesamtumfang statt. Daten wie beispielsweise Preislisten und Beschreibungstexte sind damit nicht mehr zu verarbeiten. Deshalb wurde eine weitere Variante entworfen.

Sie greift nicht mehr auf *explode* zurück und versucht eine serielle⁷⁸ Datenverarbeitung. Das neue Verfahren arbeitet mit Positionszeigern und der Funktion *substr*⁷⁹. Ziel ist es, mehrere Zeiger zu setzen und den Inhalt dazwischen auszuschneiden. Es ist bildlich mit einem Kuchen zu vergleichen aus dem mehrere kleine Stücke entnommen werden. Der Zeilenumbruch stellt auch für dieses Verfahren einen wichtigen Bestandteil dar. Während der erste Zeiger auf der Startposition Null verweilt, gibt der zweite Zeiger die Position des ersten Zeilenumbruchs wieder. Alle Zeichen die zwischen den beiden Zeigern liegen stellen eine Zeile im Dokument dar. Nachfolgende Zeilenlängen werden durch die Differenz aus End- und Startposition berechnet. Der Zeiger für die folgende Startposition wird nach der aktuellen Endposition gesetzt. Diese kann nun mit Hilfe der PHP Funktion *substr* extrahiert werden. Sie benötigt eine Startposition und eine Längenangabe, also Anzahl zu extrahierender Zeichen, und ist deshalb für diese Aufgabe wunderbar geeignet. Aus der so ermittelten Zeichenkette sind im Anschluss die Spaltenangaben zu gewinnen. Leider funktioniert auch hier die *explode* Funktion nicht. Der Umweg über *substr* ist nötig. Die Verfahrensweise ähnelt dem bereits dargelegten. Statt dem Zeilenumbruch dient der Spaltentrenner als Orientierung. Alle ermittelten Werte werden in gleicher Art und Weise wie in der ersten Variante beschrieben in einer neuen CSV Datei erfasst.

Listing 8.2: Datenkonvertierung Methode 2

```

1  {
2  ...
3  while ($pos !== false) {
4      //ermitteln Positon an der ;de; auftaucht.
5      //Beginn der Suche ab Position im letzten Parameter
6      $pos = strpos($contents, ';de;', $zeigerpos);
7      if ($pos !== false) {
8          //ermitteln Zeilenumbruch ab der durch $pos gezeigten Position-10,

```

⁷⁸ Die einzelnen Zeilen werden dabei nacheinander verarbeitet und aufbereitet, während in der vorherigen Methode erst eine komplette Zerteilung und dann erst die Aufbereitung stattfindet.

⁷⁹ Achour 2009, Funktionsreferenz - Textverarbeitung - Zeichenketten - String-Funktionen - substr

```

9      //da vor ;de; noch ein 10 Stelliger Objektcode kommt
10     $pos2 = strpos($contents, "\n", $pos);
11 }
12 //ermittlung der Stringlänge von ;de; bis zum \n +12
13 //Zeichen wegen dem Ferienh. Objektcode
14 $laenge = $pos2-$pos;
15 //ausschneiden eines Strings aus $contents
16 //beginnend bei $pos-12 schritte $laenge Zeichen lang
17 $zeile = substr($contents, $pos+4, $laenge-5);
18 $objektcode1 = substr($contents, $pos-16, 16);
19 $objektcode2 = explode("\n", $objektcode1);
20 $objektcode3 = explode(".", $objektcode2[1]);
21 $hauscode = $objektcode3[0].'.'. $objektcode3[1];
22 $zeigerpos = $pos+1;
23 //Anzahl der Arrays zählen, sind sie kleiner
24 //der Mindestanzahl werden sie nicht übernommen!
25 if (strlen($hauscode) > 3) {
26     $hauscode = trim($hauscode);
27     $zeile = trim($zeile);
28     $eintragen = $trenner2.$anbieter.$trenner.$hauscode.
29     $trenner.'.de'. $trenner.$zeile.$trenner2."\n";
30     fwrite($handle_writefile, $eintragen);
31 }
32 }
33 ...
34 }

```

Leider stößt auch diese Variante an ihre Grenzen. Sind die Dateien zu groß, scheitert bereits das Einlesen durch *fread*⁸⁰.

Hier setzt die dritte Variante an. Ihr Ziel ist es, aus der physisch vorhandenen Datei immer nur Teilstücke zu lesen. PHP Stellt die Funktion *fseek*⁸¹ zur Steuerung des Dateizeigers bereit. Ein Auslesen muss dadurch nicht mehr zwangsläufig vom ersten Byte aus stattfinden. Dadurch entfällt ebenfalls der in Variante zwei gezeigte Ansatz, da nun der Umfang an zu verarbeitenden Daten beliebig wählbar ist. Weitere Vorteile liegen in der nun flexibel gestaltbaren Größe der Ausgabedateien. Ihre Größe kann begrenzt und die Datenmenge über mehrere Dateien verteilt werden. Für die Erfassung in der Datenbank lösen sich weitere Probleme auf. Erstellte Dateien waren mitunter zu umfangreich, um sie einlesen zu können. Hier kam es ebenfalls zu Speicherproblemen. Die Funktion *fread* bietet die Option eine fest definierte Anzahl an Bytes aus einer Datei zu lesen. Es ist sogar Pflicht einen Wert anzugeben. In den vorherigen Fällen war die Grundlage für den Wert die Größe der jeweiligen Datei. Die Aufgabe von *fread* ist es nun den Inhalt der Datei in eine Variable zu schreiben. Wenn der Datenumfang in den dreistelligen Megabyte Bereich geht, beansprucht diese Variable natürlich entsprechend Arbeitsspeicher. Durch Verarbeitungsschritte verdoppelt und verdreifacht sich die Speichernutzung noch einmal. Demnach ist es sinnvoll immer nur Teile einer Datei einzulesen, eine Verarbeitung durchzuführen und anschließend mit dem nächsten Teilstück fortzufahren. Aufgrund dieses Ideenansatzes wurde nach einer Methode gesucht um das Vorhaben zu realisieren. Hier erweist sich *fseek* als ideal. Es hat die Aufgabe, den Dateizeiger, in diesem Fall die Startposition, ab der aus einer Datei gelesen werden soll, zu setzen. Hierdurch wird das Lesen von Teilstücken einer Datei möglich. Der Umfang jedes Teils wurde auf 10MiB festgelegt. Die weiteren Verarbeitungsschritte gleichen sich derer aus Methode 1. Ein

⁸⁰ Achour 2009, Funktionsreferenz - Dateisystemrelevante Erweiterungen - Dateisystem - fread

⁸¹ Achour 2009, Funktionsreferenz - Dateisystemrelevante Erweiterungen - Dateisystem - fseek

kleines Problem bleibt bei dieser Lösung jedoch bestehen. Durch die feste Angabe einer Lese-Größe kommt es zur Zerstückelung von Datensätzen, die sich jeweils am Ende und am Anfang eines Teilbereichs befinden. Während der unvollständige Datensatz des aktuellen Teilbereichs von der Verarbeitung ausgeklammert wird, muss er als vollständiger Satz im nächsten Teilbereich wieder auftauchen. Als Startwert für *fseek* dient eigentlich eine feste Größe von 10MiB auf die nach jedem Teil weitere 10MiB addiert werden. Um den unvollständigen Datensatz erfassen zu können, ist ein Rücklauf um mehrere Bytes erforderlich. Deshalb folgt mit Hilfe von *strpos*⁸² eine Auswertung aller Zeilenumbrüche des aktuellen Teilbereichs. Es ist davon auszugehen, dass nach dem letzten Umbruch ein unvollständiger Datensatz folgt. Anhand der fest definierten Größe für Teilbereiche und der Position des Letzten Umbruchs lässt sich die Anzahl an Bytes ermitteln, welche den unvollständigen Datensatz umfassen. Diese Größe wird von dem festgelegten Wert abgezogen und als Startwert für den nächsten Bereich gesetzt, so dass der letzte Datensatz aus dem ersten Teil im zweiten Teil vollständig auftaucht.

Listing 8.3: Datenkonvertierung Methode 3

```

1 {
2   ...
3   /* Datei öffnen und lesen */
4   $handle = fopen($objektverz.$dateiname, "r");
5   // Wiederholend 10MiB lesen und in Dateien schreiben bis Dateiende
6   while ($abbruch != true) {
7     $bytezeiger = $bytezeiger+$stripbytes;
8     // jeweils 10MiB lesen
9     $contents = fread($handle, $stripbytes);
10    // Daten verarbeiten
11    [...]
12    // Daten in Datenbank eintragen
13    [...]
14    // Position des letzten Zeilenumbruchs bestimmen
15    // und Startpunkt für nächste Datei setzen
16    $startpunkt = 0;
17    $pos = 0;
18    while ($pos != false) {
19      $sende = $pos;
20      $pos = strpos($contents, "\n", $startpunkt);
21      $startpunkt = $pos+1;
22    }
23    $rewind = $stripbytes-$sende+1;
24    fseek($handle, $bytezeiger-$rewind);
25    // Überprüfung Dateigröße zuletzt erzeugter Datei
26    // wenn 0, dann Schleife beenden da alle Daten ausgewertet
27    $bytes = filesize($dateiwrite);
28    if ($bytes < 1) {
29      $abbruch = true;
30    }
31  }
32  ...
33 }
```

⁸² Achour 2009, Funktionsreferenz - Textverarbeitung - Zeichenketten - String-Funktionen - *strpos*

8.3.2. Bearbeitung von Datensätzen im XML Format

Für die Verarbeitung von XML stellt PHP ab Version 5 die *SimpleXML* Funktionen bereit. Die vorliegenden Dateien werden mit Hilfe von *simplexml_load_file*⁸³ aus einer Datei gelesen, in ein Objekt⁸⁴ umgewandelt und liegen anschließend in einer selbst gewählten Variable vor. Um Sicherheit zu gewähren, prüft die PHP Funktion *file_exists*⁸⁵ im Vorfeld ob die XML Datei existiert. War der Vorgang erfolgreich steht das XML Objekt im Script zur Verfügung. Da der Aufbau von XML einer Baumstruktur gleicht, kann für die Verarbeitung einzelner Knoten im Dokument die Foreach Schleife angewendet werden. Sie gibt alle Elemente eines Knotens aus bis das letzte Element erreicht wurde. Vor der Verarbeitung bestimmter Knoten wird eine Datei geöffnet und später mit Daten aus der geparsten⁸⁶ XML Datei befüllt. Foreach fragt den im Script definierten Knoten auf seine Elemente ab. Innerhalb der Schleife werden Informationen die in den Elementen verkapselt sind und eventuell vorhandene Elementeigenschaften auf Variablen verteilt. Die weitere Vorgehensweise ähnelt der aus dem Kapitel *Bearbeitung von Datensätzen im CSV Format*. Aus den Werten der Variablen lässt sich eine CSV konforme Zeichenkette bilden und anschließend in eine neue Datei schreiben. Diese Datei wird in die Datenbank übertragen.

Listing 8.4: Datenkonvertierung XML

```

1 {
2     ...
3     /* gekürztes Beispiel XML Konvertierung, Hausbeschreibung */
4     $xml = simplexml_load_file($dateiload);
5
6     //Jedes XML Objekt namens "objekt" abarbeiten
7     foreach($xml->objekt as $fobjekt) {
8         $catidvor = $fobjekt->objektID;
9         if (!preg_match($slashed.$catidvor.$slashed, $catidvor2)) {
10             // geht nacheinander alle Text Childs
11             // des jeweiligen Objekt Knotens durch
12             foreach ($fobjekt->text AS $beschrtext) {
13                 if ($beschrtext['topic'] == 'beschreibung' AND $beschrtext['nr'] == '1') {
14                     $desc-haus1 = (string)$beschrtext;
15                 }
16                 if ($beschrtext['topic'] == 'entfernungen' AND $beschrtext['nr'] == '1') {
17                     $desc-entfernung1 = (string)$beschrtext;
18                 }
19             }
20             $eintragen = $strenner2.$anbieter.$strenner.$catidvor.
21             $strenner.'de'.$strenner.$desc-haus1.'<br><br>'
22             .$desc-entfernung1.$strenner2.'"\\n";
23             fwrite($handle_writefile, $eintragen);
24         }
25         $catidvor2 = $fobjekt->objektID;
26     }
27     ...
28 }

```

⁸³ Achour 2009, Funktionsreferenz - XML-Manipulation - SimpleXML - SimpleXML-Funktionen - simplexml_load_file

⁸⁴ Kannengiesser 2004, 389ff

⁸⁵ Achour 2009, Funktionsreferenz - Dateisystemrelevante Erweiterungen - Dateisystem - file_exists

⁸⁶ Wikipedia Parser, Stand: 24. Januar 2009, aufgerufen am 17. April 2009

Eine Kleinigkeit ist bei der Verarbeitung von XML Dokumenten zu beachten. Elemente stellen einzelne Objekte dar. Für die korrekte Übertragung als Zeichenkette muss den Objekten die Typdefinition (*string*)⁸⁷ vorangestellt sein. Aus der praktischen Arbeit ergaben sich durch weglassen zwar keine unerwarteten Auswirkungen, der Korrektheit halber sollten sie dennoch gesetzt sein. Die Verarbeitung von XML Dokumenten führte in der Anwendung zu keiner Überbeanspruchung des verfügbaren Speichers. Diese Variante bildet das Grundprinzip für die Erfassung und Verarbeitung der Daten. Je nach Dokument und enthaltener Informationen sind weitere Schritte nötig.

8.3.3. Konvertierung von Wolters Datenbeständen

Wolters beispielsweise stellt seine Daten im XML Format bereit. In diesem Abschnitt geht es um Besonderheiten, die bei der Konvertierung zu berücksichtigen sind.

Die Verarbeitung der Merkmale erfordert z.B. das Laden von zwei XML Dateien. Es handelt sich neben dem Dokument für Merkmale um die Stammdaten Datei. Sie enthält Informationen über die max. zulässige Personenanzahl der Objekte, welche in den Merkmalen nicht aufgeführt ist. Im Script steht die Verarbeitung der Stammdaten an erster Stelle. Aus den vorliegenden Daten wird ein Array erzeugt dass Objektkennung, Kategorie und Anzahl maximal zulässiger Personen und Kinder enthält.

Erst nach diesem Schritt steht die eigentliche Konvertierung der Merkmalsdatei an, wobei im ersten Schritt die Personen- und Kinderanzahl aus dem Array zugewiesen wird. Dazu erfolgt über die PHP Funktion *preg_match*⁸⁸ ein Vergleich der Objektkennung und seiner Kategorie mit den entsprechenden Feldern aus dem Array, dass die max. Belegungsanzahl enthält. Stimmen sie überein werden die Werte übernommen.

In der Datenbank, eine Erläuterung und Definition findet im Kapitel 6 auf Seite 35 statt, liegen Merkmale als Null oder Eins Wert codiert vor. Wolters liefert jedoch als Merkmalsangabe die Beschreibung des Merkmals selbst. Existiert im Objekt ein Whirl-Pool ist dieser als „Whirl-Pool“ angegeben. Eine Auflistung aller möglichen Merkmale und die Kennzeichnung beispielsweise als „ja“ oder „nein“ findet nicht statt. Im Vorfeld müssen also alle Merkmale, die Wolters anbietet, bekannt sein. Der Anbieter stellt dazu Dokumente zur Verfügung⁸⁹, die entsprechende Informationen bereit stellen. Laut Information existieren 35 unterschiedliche Merkmale.

Aus diesen Informationen lässt sich eine Schleife entwickeln, die sich so oft aufruft wie es max. Merkmale pro Objekt gibt und versucht, jedes Merkmal in eine „1/0“ Codierung zu konvertieren. Innerhalb der Schleife sind alle 35 Merkmale als separate Bedingung aufgeführt und in einem Array das entsprechende Feld auf Null gesetzt. Jedes Merkmal wird mit allen Merkmalen in der Schleife verglichen. Liegt eine Übereinstimmung vor, ändert sich der Wert im zugehörigen Feld auf Eins. Als letzter Schritt folgt wie bekannt die Bildung einer CSV Konformen Zeichenkette, die einen Eintrag in der Merkmalstabelle widerspiegelt.

⁸⁷ Achour 2009, Sprachreferenz - Typen

⁸⁸ Achour 2009, Funktionsreferenz - Textverarbeitung - PCRE-Funktionen - *preg_match*

⁸⁹ Es handelt sich dabei um elektronische Dokumente im PDF Format, die auf dem FTP Server des Anbieters zum Download bereit stehen

Listing 8.5: Datenkonvertierung Merkmale Wolters

```

1 }
2 ...
3 /* Auszug, Konvertierung Merkmale */
4 {
5     ...
6 }
7 // Zuweisen Anzahl Personen und Kinder zu Objekt
8 if (preg_match("/\b$objreferenz\b/i", $personen_array[$i][0])
9 AND preg_match("/\b$objektschluesel\b/i", $personen_array[$i][1])) {
10     $personen_anz = $personen_array[$i][2];
11     $kinder_anz = $personen_array[$i][3];
12 }
13 $i++;
14
15 // fluschen der vorherigen Arrayeinträge, 35 = Anzahl aller möglichen Merkmale
16 for ($i = 0; $i < 35; $i++) {
17     $merkmale_sammeln[$i] = 0;
18 }
19 // Datengewinnung aus den Merkmalsknoten
20 for ($i = 0; $i < $merkmalanzahl; $i++) {
21     $merkmal = utf8_decode((string)$fmerkmale->merkmal[$i]);
22     $merkmal_wert = (string)$fmerkmale->merkmal[$i]['wert'];
23     /* *Notwendige Fallunterscheidung da Position der Merkmale variiert* */
24     if ($merkmal == 'Anlage') {
25         $merkmale_sammeln[0] = 1;
26     }
27
28     if ($merkmal == 'Anz._Haustiere') {
29         $merkmale_sammeln[1] = $merkmal_wert;
30     }
31
32     if ($merkmal == 'Anzahl_Schlafräume') {
33         $merkmale_sammeln[2] = $merkmal_wert;
34     }
35
36     {
37         ...
38     }
39
40 }
41 ...
42 }

```

Preisdaten sind von Wolters mit einem Statusbyte versehen. Das Konvertierungsscript für Preise beachtet diese Information. Einträge mit der Kennung I für Inaktiv finden deshalb nicht ihren Weg in die Datenbank. Pro Objekt gibt es vier Preise, da ein Objekt für 1 - 4 Wochen buchbar ist. So werden pro Schleifenaufruf statt einem - gleich vier neue Zeilen in die neue Datei geschrieben.

Beschreibungstexte gliedert der Anbieter in mehrere Kategorien. Unterschieden wird in die eigentliche Beschreibung, Rabatte, Zusatzleistungen, Kategoriebeschreibung und Entfernungsangaben zu wichtigen Einrichtungen. Die Datenbank besitzt jedoch nur Entitäten für die Erfassung von Haus- und Wohnungsbeschreibung. Ein Script führt die Extraktion der Haus- das andere die der Wohnungsbeschreibungen durch. Ersteres fasst während der Konvertierung Beschreibung und Entfernungsangaben zusammen. Im Script für die Zimmerbeschreibung folgt eine Zusammenfassung von Kategoriebeschreibung, Zusatzleistungen und Rabattmöglichkeiten.

8.3.4. Konvertierung von Stammdaten

Voraussetzung für eine einwandfrei funktionierende Suche ist die eindeutige Referenzierung aller Objekte. Sie beginnt mit der Erfassung der Lagedaten aller Objekte. Hauptziel des Users ist die Reise in eine bestimmte Gegend. Die Erfassung mehrerer Anbieterdaten in einem System stellt sich an dieser Stelle als nicht trivial heraus, da jeder Anbieter abweichende Codierungen und Namen benutzt. An dieser Stelle muss ein System geschaffen werden was in der Lage ist die Daten so abzugleichen, dass Objekte unterschiedlicher Anbieter korrekt auf das entsprechende Land, die Region und den passenden Ort referenziert werden.

Im ersten Schritt steht der Entwurf einer eigenen Codierung an. Während der Datenerfassung folgt eine Umwandlung der jeweiligen Anbietercodierung auf das eigene System. Kriterien sind die Eindeutigkeit der Codes in Bezug auf Format und Inhalt und eine leichte Verständlichkeit um die spätere Nutzung zu vereinfachen. Unter die Codierung fallen Länder, Regionen und Orte. Während für die Codierung der Länder die Abkürzung des Landesnamen genutzt wird, sind Regionen und Orte als Zahlenfolge codiert. Regionen bestehen aus einer vierstelligen- und Orte aus einer sieben-stelligen Zahlenfolge. Die Ortscodierung enthält dabei den vierstelligen Schlüssel der zugehörigen Region, gefolgt von einer dreistelligen Angabe für den Ort. Pro Region sind 999 Orte möglich. Für Regionen besteht eine ähnliche Einteilung. Die ersten beiden Ziffern geben Aufschluss über das Land, die letzten beiden Ziffern über die Region. Pro Land sind 99 Regionen möglich. Aufgrund dieser Einteilung ist eine genaue Identifizierung und Abgrenzung möglich. Zum einen kann der geübte Betrachter den jeweiligen Orten sofort die entsprechende Region zuweisen, zum anderen sind bsp. gleichlautende Bezeichnungen von Orten unterschiedlicher Regionen kein Problem. Später kommt dieses verschachtelte System der automatisierten Identifizierung alleine anhand von Namen der Länder, Orte und Regionen zugute.

Es stellt einen Teil eines größeren Systems dar. Für jeden Anbieter existiert in der Datenbank eine Referenztabelle, die zu jedem Land, Region und Ort sowohl Anbietercodierung als auch interne Codierung enthält. Da jeder Anbieter seinen Datenbestand ständig erweitert, muss vor der Stammdatenkonvertierung ein Update auf die Referenztabelle durchgeführt werden. Für das Update stehen weitere drei Tabellen zur Verfügung. Sie enthalten Bezeichnung und Referenzcodes der Länder, Regionen und Orte. Wichtig ist eine Referenztabelle für Länder die bereits alle Länder mit entsprechender Codierung im Vorhinein enthält. Sie stellt die oberste Ebene für Referenzierungsaufgaben dar und kann über keine höhere Instanz korrekt erweitert werden. Fehlende Einträge führen zu fehlerhaften Ergänzungen in den darunter liegenden Ebenen für Regionen und Orte. Eine vollständige Liste mit Ländern im Vorfeld per Hand anzulegen stellt kein Problem dar, denn ihre Anzahl ist überschaubar.

Bei einem Update der Referenztabellen des entsprechenden Anbieters läuft folgender Prozess ab. Jedes einzelne Objekt aus den heruntergeladenen Stammdaten wird mit der Anbieter-Referenztabelle verglichen. Existiert eine Referenz für das Land, die Region und den Ort ist alles ok und das nächste Objekt wird verglichen. Sollte eine der drei Referenzierungen nicht erfolgreich sein, beginnt ein Vergleichsalgorithmus seine Arbeit.

←T→			IDLand	Land_kurz	Land
<input type="checkbox"/>			AT	AT	Österreich
<input type="checkbox"/>			AU	AU	Australien
<input type="checkbox"/>			BE	BE	Belgien
<input type="checkbox"/>			CA	CA	Kanada
<input type="checkbox"/>			CH	CH	Schweiz
<input type="checkbox"/>			CZ	CZ	Tschechische Rep
<input type="checkbox"/>			DE	DE	Deutschland
<input type="checkbox"/>			DK	DK	Dänemark
<input type="checkbox"/>			ES	ES	Spanien
<input type="checkbox"/>			FR	FR	Frankreich
<input type="checkbox"/>			GB	GB	United Kingdom
<input type="checkbox"/>			GR	GR	Griechenland
<input type="checkbox"/>			HR	HR	Kroatien
<input type="checkbox"/>			HU	HU	Ungarn
<input type="checkbox"/>			IE	IE	Irland
<input type="checkbox"/>			CH	IH	Schweiz
<input type="checkbox"/>			IT	IT	Italien

Abbildung 8.2.: Auszug aus der Tabelle ref_land

Bestandteile sind die Tabellen mit der internen Landes-, Regions- und Ortsreferenzierung und die Referenztabelle des jeweiligen Anbieters. Im ersten Schritt sucht der Algorithmus in der Anbieter-Referenztabelle nach der Codierung des Landes. Gemeint ist damit die Codierung die der Anbieter verwendet. Wird er fündig, speichert er den zugehörigen internen Code und sucht als nächstes nach der Regionscodierung. Ist diese ebenfalls vorhanden, folgt der Versuch den passenden Ort aus der internen Ortsreferenzierung zu ermitteln. Dazu sind die internen Codes von Land und Region nötig. Nur durch sie kann, bei mehrfachem Auftreten ein und desselben Ortsnamens, der korrekte Ort zugewiesen werden. Die Suche nach einem Ort teilt sich in zwei Varianten auf. Variante 1 führt eine Suche mit Hilfe von Wildcards⁹⁰ am Beginn des Namens aus. Treten dadurch mehrere Ergebnisse auf, folgt eine genauere Suche ohne Wildcards. Variante 1 soll evtl. verbliebene Leerzeichen am Beginn eines Namens berücksichtigen und vermeiden, dass ein und derselbe Ort mehrmals in der Datenbank landet. Gibt auch die genauere Suche mehrere Ergebnisse aus, wird der letztgenannte Eintrag herangezogen. Die weiteren Schritte sind in beiden Varianten gleich. Der Ablauf ist in Kapitel C.10 auf Seite 156 ersichtlich.

Sollte es einen passenden Eintrag geben wird dessen ID zwischengespeichert und übergeben. Im Anschluß folgt die Ergänzung in der Referenztabelle für den jeweiligen Anbieter. Ist auch in der entsprechenden Referenztabelle für Region oder Ort, je nach Fall, kein Eintrag zu finden, folgt ein Abruf aller zum Land bzw. Region gehöriger Regionen bzw. Orte und ihre ID. Die Sortierung orientiert sich an der ID und ist aufsteigend.

⁹⁰ Wildcards stellen einen Platzhalter für ein oder mehrere Buchstaben dar. Ihr Prinzip entspricht dem eines Jokers in einem Kartenspiel. Ein Stern ist Platzhalter für eine beliebige Anzahl an Zeichen. Das Fragezeichen dagegen steht als Platzhalter für ein einzelnes Zeichen.

Hintergrund dieser Aktion ist es, eine Grundlage zu schaffen um den nicht existierenden Datensatz zu ergänzen. Wichtig ist es, an die höchste vergebene ID anzuknüpfen. Sie bildet die Grundlage für eine fehlerfreie Nummerierung des folgenden Datensatzes. Der Wert der momentan höchsten vergebenen ID wird um einen Zähler erhöht und bildet die ID für die neue Region oder den neuen Ort.

Listing 8.6: Datenupdate Regionstabelle

```

1 }
2 ...
3 if ($zeilen_fh_region < 1) {
4   $query_fh_region_neu = "SELECT_*_FROM_ref_region
5   WHERE_LandID=_ '$land_ref' ORDER_BY_IDRegion";
6   $aufruf_fh_region_neu = mysql_query($query_fh_region_neu);
7   $regionen = mysql_num_rows($aufruf_fh_region_neu);
8   $neuregionen = 0;
9   if ($regionen < 1) {
10    $neuregionen = 1;
11    $query_fh_region_neu = "SELECT_IDRegion FROM_fh_region
12    ORDER_BY_IDRegion";
13    $aufruf_fh_region_neu = mysql_query($query_fh_region_neu);
14  }
15  while ($line_aufruf_fh_region = mysql_fetch_array(
16    $aufruf_fh_region_neu, MYSQL_ASSOC)) {
17    $region_last = $line_aufruf_fh_region['IDRegion'];
18  }
19  if ($neuregionen == 1) {
20    $region_last = substr($region_last,0,2);
21    $region_last = $region_last+1;
22    $region_last = $region_last.'01';
23  }
24  else {
25    $region_last = $region_last+1;
26  }
27  $insert_fh_region = "INSERT INTO_ref_region_(IDRegion, LandID, Region)
28  VALUES(' $region_last ', '$land_ref ', '$fh_region_name ')" ;
29  }
30  ...
31 }

```

Zu erwähnen ist, dass die Neueintragung einer Region gleichzeitig einen Ersteintrag in der Ortstabelle auslöst. Es ist ein logischer Schritt, denn für die entsprechende Region können noch keine Orte vorhanden sein. So lässt sich der Ort für diesen Datensatz sofort auslesen und muss nicht erst ab der Ortssuche ergänzt werden. Wenn der Eintrag in die entsprechende Referenztablette erfolgreich war, steht eine Ergänzung in der Referenztablette für den Anbieter an. Dieser Prozess wiederholt sich so lange wie es neue, nicht in der Referenztablette des Anbieters gelistete Eintragungen gibt. Er schließt mit der Auswertung des letzten Datensatzes der Stammdaten ab. Da nun alle Tabellen ein Update erfahren haben, kann die eigentliche Konvertierung der Stammdaten starten. (Siehe Anhang Stammdatenscript)

8.3.5. Konvertierung Vakanzen von Interhome

Wie bereits im Kapitel 6 erläutert sind die Belegungs-, Übergabeinformationen und Mindestaufenthalt nach Monaten erfasst. Interhome liefert die Belegungen als Zeichenkette mit einem Umfang von bis zu einem Jahr. Neben der Belegung ist sowohl ein Datum ab

welchem die Vakanzen gültig sind, als auch ein Datum bis wann Vakanzen gelten, vorhanden. Meist liegt der Starttag irgendwo im Aktuellen Monat. Alle Tage die davor liegen müssen ergänzt werden, um den Monat zu vervollständigen. Also ist im ersten Schritt eine Ermittlung der Anzahl an Tagen des Monats in dem die Belegung beginnt, sowie der im Datum genannte Tag Voraussetzung. PHP bietet für solche Fälle die Funktionen *date* und *mktime*⁹¹. *Mktime* wandelt das Startdatum in einen Unix Timestamp⁹², der im Anschluss an die Funktion *date* übergeben wird. *Date* bietet viele Formen der Datenausgabe an. Es können aus einem Datum der Wochentag in numerischer Form oder als ganzes Wort, ein entsprechender Monat, Woche oder Anzahl an Tagen des Monats ausgegeben werden. Im geschilderten Fall liegt das Augenmerk auf letztgenanntem. Ist die Anzahl an Tagen des gesuchten Monats bekannt, lassen sich über einige Rechenschritte die fehlenden und bereits vorhandenen Einträge des Monats ermitteln.

Über eine For Schleife werden alle fehlenden Tage mit Werten ergänzt. Da ihr Belegungstatus unbekannt ist und sie in der Vergangenheit liegen, kommen nur Werte in Betracht, die den Zeitraum als belegt ausgeben. Für alle übrigen Tage sind Daten vorhanden. Gleichzeitig lässt sich so in der von Interhome gelieferten Zeichenkette der Beginn des Folgemonats ermitteln. Ab dem Folgemonat stellt sich die Aufsplittung der Zeichenkette einfacher dar, denn nun liegen für alle Tage des Monats Daten vor. Als Startposition des Folgemonats dient die auf eine Endposition des aktuellen Monats folgende Position in der Zeichenkette. Die Ermittlung der Anzahl an Tagen pro Monat ist für jeden Monat notwendige Bedingung. Andernfalls lässt sich keine korrekte Auftrennung der Zeichenkette realisieren.

Listing 8.7: Datenupdate Vakanzen Interhome

```

1 {
2     ...
3     //1. Monat
4     //-1 da $tag bereits erster Tag eines Eintrages ist
5     for ($k=0; $k < $tag-1; $k++) {
6         $vakanzstring = $vakanzstring.'N';
7         $uebergabestring = $uebergabestring.'X';
8         $minstaystring = $minstaystring.'0';
9     }
10    $vakanzstring = $vakanzstring.substr($vakanz,0,$resttage);
11    $uebergabestring = $uebergabestring.substr($uebergabe,0,$resttage);
12    $minstaystring = $minstaystring.substr($minstay,0,$resttage);
13    $datum = $jahr.'-'. $monat.'-01';
14    $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.
15    $strenner.$kategorie.$strenner.$datum.$strenner.$vakanzstring.
16    $strenner.$uebergabestring.$strenner.$minstaystring.$strenner2."\\n";
17    fwrite($handle_writefile, $eintragen);
18    $startpos = $resttage;
19    // Alle weiteren Monate
20    for ($k=1; $k <= 12; $k++) {
21        $monat = $monat+1;
22        if ($monat == 13) {
23            $monat = 1;
24            $jahr = $jahr+1;
25        }
26        $t_mon = date("t", mktime(0, 0, 0, $monat, 1, $jahr));

```

⁹¹ Achour 2009, Funktionsreferenz - Kalender- und ereignisrelevante Erweiterungen - Datum/Uhrzeit Funktionen

⁹² Wikipedia Timestamp, Stand: 15. Februar 2009, aufgerufen am 20. April 2009

```

27 //Achtung Substring beginnt bei Null zu zählen deshalb ist
28 //startpos automatisch Folgetag nach letztem Tag im vorherigen Feld
29 $vakanzstring = substr($vakanz,$startpos,$t_mon);
30 $uebergabestring = substr($uebergabe,$startpos,$t_mon);
31 $minstaystring = substr($minstay,$startpos,$t_mon);
32 if (strlen($monat) < 2) {
33     $monat = '0'.$monat;
34 }
35 $datum = $jahr.'-'.$monat.'-01';
36 $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.
37 $trenner.$kategorie.$trenner.$datum.$trenner.$vakanzstring.
38 $trenner.$uebergabestring.$trenner.$minstaystring.$trenner2."\n";
39 fwrite($handle_writefile, $eintragen);
40 $startpos = $startpos+$t_mon;
41 }
42 ...
43 }

```

8.3.6. Übernahme Privatangebote aus aktuellem in das neue System

Einen Sonderfall in der Verarbeitung stellt die Übernahme der Privatangebote aus dem momentan existierenden System dar. Die Informationen liegen nur in der dort benutzten Datenbank vor. Exportdateien gibt es nicht. So ist die Möglichkeit gegeben, Exportdateien nach den Spezifikationen der neuen Datenbank zu erstellen. Eine Erstellung der Exportdateien findet auf dem Ursprungsserver statt. Der neue Server holt sie wie gewohnt über das FTP Protokoll von dort ab.

Der Export aller Daten wird durch ein Script realisiert. Dies betrifft Stammdaten, Texte, Merkmale, Vakanzen und Bilder. Die Informationen werden in Dateien mit entsprechender Bezeichnung gespeichert.

Das Script checkt die Dateien auf Beschreibbarkeit. Sollte der Schreibvorgang erfolgreich sein, werden die alten Inhalte gelöscht. Innerhalb einer While Schleife liest das Script aus der Tabelle *hotels* den Objektcode. Einen Objektcode in der Form wie sie die neue Datenbank vorsieht ist nicht vorhanden. Das System besteht aus einer fortwährenden Nummerierung der Objekte. Um eine kompatible Darstellung zu erreichen, wird die einfache Nummerierung bearbeitet. Der neue Code ist zehnstellig und besteht aus einem vierstelligen Buchstabenkürzel, gefolgt von einer sechsstelligen Nummer. In dieser Nummer ist die vormals verwendete ID wiederzufinden. In der ursprünglichen Datenbank liegt ebenfalls eine Einteilung in Haustypen vor. Sie dient in der neuen Form als Kategorieangabe. Aus den Typenbezeichnungen werden zweistellige Kategoriekürzel gebildet. Beispielsweise ist ein Haus als „HA“, eine Feriendorf als „FD“ wiederzufinden. Gibt es keine Angabe in der Kategorie, wird die Abkürzung „KA“ als Kategorie benutzt. Alles was an Beschreibungen und Besonderheiten zum Objekt in der Datenbank gespeichert ist, kommt als ein gesamter Text in die spätere CSV Datei.

Listing 8.8: Generierung Referenzcode - Übernahme Beschreibungstexte

```

1 {
2     ...
3     //k ist die Anzahl an einzufügenden Nullen
4     for ($k = 0; $k < $nullserie; $k++) {

```

```

5     $id_hotel_new = $id_hotel_new.'0';
6 }
7 $id_hotel_new = 'ONFH'. $id_hotel_new. $id_hotel;
8 /* Objekttyp -> Objektkategorie */
9 switch ($objekttyp) {
10     case 'Haus':
11         $objekttyp = 'HA';
12         break;
13     case 'Wohnung':
14         $objekttyp = 'WH';
15         break;
16     case 'Feriendorf':
17         $objekttyp = 'FD';
18         break;
19     case '':
20         $objekttyp = 'KA';
21         break;
22 }
23 $text = $beschreibung."\\n".$besonderheiten;
24 $hausbeschreibung_eintragen = $strenner2.$idanbieter.$strenner.
25 $id_hotel_new.$strenner.'de'.$strenner.$text.$strenner2."\\n";
26 fwrite($handle_writefile_texte, $hausbeschreibung_eintragen);
27 ...
28 }

```

Die Auswertung der Merkmale ist aufwändiger. Im ursprünglichen System ist jedem Merkmal innerhalb der Tabelle *hotels* eine eigene Spalte zugeordnet. Angaben über max. Belegungsanzahl, Entfernungen oder Zimmeranzahl können zwar ohne Umwege übernommen werden, Angaben mit „ja“ oder „nein“ nicht. Sie sind als Enumeration⁹³ mit den Auswahlmöglichkeiten „ja“ und „nein“ versehen. Jeder dieser Merkmalsangaben muss deshalb auf seine Angabe überprüft werden. Je nach Fall steht eine Konvertierung zu „0“ oder „1“ an, wobei „0“ für nein oder keine Angabe steht. Für einige Merkmale folgt eine Zusammenfassung, etwa - wenn es sich um Dinge wie „schwimmbad-innen“ und „schwimmbad-aussen“ handelt. Hier reicht es aus, wenn einer der beiden Werte eine positive Angabe aufweist, um eine „1“ zu setzen. Jedes Zimmer ist in diesem System extra ausgewiesen. An dieser Stelle wird die Summe aus allen aufgeführten Zimmern gebildet.

Listing 8.9: Konvertierung und Export von Merkmalen

```

1 {
2     ...
3     /* — Privathäuser Merkmale — */
4     $betten = $abruf_line['betten'];
5     $personen = $abruf_line['kapazitaeten-erwachsene'];
6     $kinder = $abruf_line['kapazitaeten-kinder'];
7     $schlafzimmer = $abruf_line['anzahlzimmer-schlafzimmer'];
8
9     if ($abruf_line['haustiere'] == 'ja') {
10         $haustiere = 1;
11     }
12     else { $haustiere = 0; }
13     if ($abruf_line['wellness'] == 'ja') {
14         $wellness = 1;
15     }
16     else { $wellness = 0; }
17     if ($abruf_line['pool'] == 'ja') {
18         $pool = 1;

```

⁹³ EDM2 Enumeration, Stand: 2. Dezember 2004, aufgerufen am 5. Juni 2009


```

19 }
20 else { $pool = 0; }
21 ...
22 if ($abruf_line['ausstattung-aussenbereich-garage'] == 'ja'
23 OR $abruf_line['ausstattung-aussenbereich-tiefgarage'] == 'ja') {
24     $garage = 1;
25 }
26 else { $garage = 0; }
27
28 $zimmer = $abruf_line['anzahlzimmer-schlafzimmer-kinder']+
29 $abruf_line['anzahlzimmer-wohnzimmer']+
30 $abruf_line['anzahlzimmer-esszimmer']+
31 $abruf_line['anzahlzimmer-wohn-esszimmer']+
32 $abruf_line['anzahlzimmer-kueche'];
33 $merkmale_eintragen = $strenner2.$idanbieter.$strenner.$id_hotel.new.
34 $strenner.$objekttyp.$strenner.''. $strenner.'0'. $strenner.'0'.
35 ...
36 fwrite($handle_writefile_merkmale, $merkmale_eintragen);
37 ...
38 }

```

Jedes Objekt besitzt eine Bildergalerie. Passende Bilder liegen im Verzeichnis des Ferienobjektes vor. Objekte sind nach Ländern geordnet. Der Pfad zu den Bildern beinhaltet also mindestens zwei Verzeichnisangaben. Dem Aufruf des entsprechenden Verzeichnisses für das Land folgt ein Sprung in das Verzeichnis was nach dem Objektnamen benannt ist. Bilder befinden sich wiederum im Unterordner „Images“. Dieser Ordner existiert für jedes Objekt. Aus diesen Angaben lässt sich eine URL⁹⁴ zu den Bildern auf dem Ursprungsserver erstellen. Aus der Datenbank des genannten Servers lässt sich auch der Verzeichnisname des Landes ermitteln. Die Tabelle *a.land* enthält die nötigen Angaben. Der Verzeichnisname für das entsprechende Hotel kann analog aus der Tabelle *hotels* ausgelesen werden. Den Bildern ist als Dateiname eine ID zugeordnet. Über die Tabelle *hotels_bilderliste* lassen sich alle momentan aktiv geschalteten Bilder ermitteln. Da jede Bild ID gleichzeitig den Dateinamen darstellt, ist lediglich eine Übernahme der ID aus der Datenbank notwendig, um die zu erzeugende URL vervollständigen zu können.

Listing 8.10: Konvertierung und Export von Bildlinks

```

1 {
2     ...
3     /* — Privathäuser hausbilder — */
4     $verzeichnis2 = $abruf_line['exportverzeichnis'];
5     $idland = $abruf_line['id_land'];
6     $querylanddir = "SELECT _verzeichnis _FROM _a_land
7 _WHERE _id_land _= _'$idland' _LIMIT _1";
8     $abruflanddir = mysql_query($querylanddir);
9     while($abruf_line = mysql_fetch_array(
10 $abruflanddir, MYSQL_ASSOC)) {
11         $verzeichnis1 = $abruf_line['verzeichnis'];
12     }
13     $url = 'http://www.ferienhaus.de/ferienhaeuser/'.
14     $verzeichnis1.'/'. $verzeichnis2.'/images/';
15
16     $querybilder = "SELECT _id_bild _FROM _hotels_bilderliste
17 _WHERE _id_hotel _= _'$id_hotel'";
18     $abrufbilder = mysql_query($querybilder);
19     $k = 1;
20     while($abruf_line = mysql_fetch_array($abrufbilder, MYSQL_ASSOC)) {
21         $bild = $abruf_line['id_bild'];

```

⁹⁴ Jamsa 1996, 99ff

```

22     $bildlink = $url.$bild.'.jpg';
23
24     //echo 'Bildlink: '. $bildlink.'<br>';
25     if ($verzeichnis2 == $verzeichnis_old) {
26         $k++;
27     }
28     else {
29         $k = 1;
30     }
31     $bilder_eintragen = $strenner2.$idanbieter.$strenner.$id_hotel_new.
32     $strenner.$k.$strenner.'NULL'.$strenner.$bildlink.$strenner2."\\n";
33     fwrite($handle_writefile_bilder, $bilder_eintragen);
34     $verzeichnis_old = $verzeichnis2;
35 }
36 ...
37 }

```

Logischerweise ist auch der Export von Stammdaten wichtig. Jedes Objekt ist mit einer ID für Land und Region gekennzeichnet. Die Tabellen *a_land* und *a_region* auf dem Ursprungsserver geben Auskunft welche ID welchem Land oder Region zugeordnet ist. Für Orte gibt es keine Referenztabelle. Ortsnamen sind direkt in die Hotelangaben integriert. Aus den drei Tabellen werden die nötigen Informationen extrahiert und in der Datei *stammdaten.csv* abgelegt. Jeder Eintrag besteht aus einer Anbieterkennung, Objekt- und Kategorieangabe, Name von Land, Region und Ort. Als interne Anbieterkennung kommt übrigens die „99“ zum Einsatz. Für Interhome ist beispielsweise die „1“ reserviert.

Listing 8.11: Export der Stammdaten

```

1 {
2     ...
3     /* —— Privathäuser Stammdaten —— */
4     //IDAnbieter, ObjReferenz, Kategorie, Land, Region, Ort
5     $queryland = "SELECT name FROM a_land WHERE id_land = '$id_land' LIMIT 1";
6     $abrufland = mysql_query($queryland);
7     while($abrufl_line = mysql_fetch_array($abrufland, MYSQL_ASSOC)) {
8         $landesname = ltrim(rtrim(addslashes(utf8_encode($abrufl_line['name']))));
9     }
10
11     $queryregion = "SELECT name FROM a_region WHERE id_region = '$id_region' LIMIT 1";
12     $abruflregion = mysql_query($queryregion);
13     while($abrufl_line = mysql_fetch_array($abruflregion, MYSQL_ASSOC)) {
14         $regionsname = ltrim(rtrim(addslashes(utf8_encode($abrufl_line['name']))));
15     }
16
17     $stammdaten_eintragen = $strenner2.$idanbieter.$strenner.$id_hotel_new.
18     $strenner.$objekttyp.$strenner.$landesname.$strenner.$regionsname.
19     $strenner.$ortsname.$strenner2."\\n";
20     fwrite($handle_writefile_stammdaten, $stammdaten_eintragen);
21     //echo 'Stammdaten_EINTRAGEN: '. $stammdaten_eintragen.'<br>';
22     $i++;
23     ...
24 }

```

Eine Aufbereitung und der Export von Daten für die Belegungsübersicht bilden den letzten Teil im Script. Angaben zur Belegung finden sich in der Tabelle *hotel_belegungsuebersicht*. Die dortige Form ist allerdings nicht mit der Form aus der neuen Vakanztabelle, wie in Abschnitt 6.3.11 beschrieben, kompatibel. Deshalb ist eine Konvertierung nötig. Dazu findet eine Datenbankabfrage statt, die alle Tage ab dem aktuellen Datum ausgibt für die eine Belegung zutrifft. Im Vorfeld wird ein mehrdimensionales Array

erzeugt welches als oberste Ebene das Jahr, gefolgt von den Monaten mit den Tagen als letzte Instanz, enthält. Jeder Tag enthält ein Zeichen, dass den Status repräsentiert. Bevor die Datenbankabfrage erfolgt, präsentieren alle Werte in diesem Array den Status „Objekt zu diesem Zeitpunkt buchbar“. Anhand des Datums der Einträge aus der Datenbank kommt es im Anschluss zur Korrektur der jeweiligen Einträge auf den Status „Objekt zu diesem Zeitpunkt nicht buchbar“. Der selbe Algorithmus kommt für die Auswertung des Folgejahres zum Einsatz. Da eine kontinuierliche Bereitstellung von Daten im Umfang von einem Jahr auch zum Ende des Jahres gewährleistet sein soll, muss das Folgejahr berücksichtigt werden. Ist die Modifizierung der Werte abgeschlossen, lässt sich aus dem Array mit Hilfe von *implode*⁹⁵ eine Zeichenkette erstellen. Nach erfolgter Zusammenfassung beider Jahre folgt eine Kürzung auf 365 Zeichen, was dem Umfang eines Jahres entspricht. Jedoch gibt es noch zwei weitere Punkte auf die Rücksicht zu nehmen ist. Das betrifft Informationen über Schlüsselübergabe und Mindestaufenthaltsdauer. Ihre Aufbereitung läuft nach dem selben Prinzip ab.

Listing 8.12: Konvertierung und Export von Vakanzen

```

1 {
2   ...
3   for ($i = (int)$monat; $i <= 12; $i++) {
4     $monthdays = date("t", mktime(0, 0, 0, $i, 1, $jahr));
5     for ($k = 1; $k <= $monthdays; $k++) {
6       $vakanzen[$jahr][$i][$k] = 'Y';
7       $uebergabe[$jahr][$i][$k] = 'C';
8       $minstay[$jahr][$i][$k] = 'C';
9     }
10  }
11  // Monate ab 1. des Folgejahres, Tage ab Monatsanfang 1. Monat
12  for ($i = 1; $i <= 12; $i++) {
13    $monthdays = date("t", mktime(0, 0, 0, $i, 1, ($jahr+1)));
14    for ($k = 1; $k <= $monthdays; $k++) {
15      $vakanzen[( $jahr+1)][$i][$k] = 'Y';
16      $uebergabe[( $jahr+1)][$i][$k] = 'C';
17      $minstay[( $jahr+1)][$i][$k] = 'C';
18    }
19  }
20
21  // Vakanzen abfragen
22  $query = "SELECT_*FROM $tabelle_vak WHERE id_hotel =_ $id_hotel AND tag >=_ '$datum'";
23  $result = mysql_query($query);
24  while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
25    $hotelbeleg = $line['tag'];
26    echo $hotelbeleg.'<br>';
27    $datum_vdb = explode("-", $hotelbeleg);
28    $jahr_db = (int)$datum_vdb[0];
29    $monat_db = (int)$datum_vdb[1];
30    $tag_db = (int)$datum_vdb[2];
31
32    $vakanzen[$jahr_db][$monat_db][$tag_db] = 'N';
33    $uebergabe[$jahr_db][$monat_db][$tag_db] = 'X';
34    $minstay[$jahr_db][$monat_db][$tag_db] = 'C';
35  }
36
37  //implodieren der Tage, aktuelles Jahr
38  for ($i = (int)$monat; $i <= 12; $i++) {
39    $vak_string = implode(NULL, $vakanzen[$jahr][$i]);

```

⁹⁵ Implode stellt das Gegenstück zu Explode dar und ist deshalb im selben Kapitel wie Explode des PHP Handbuchs zu finden

```
40     $ueb_string = implode(NULL,$uebergabe[$jahr][$i]);
41     $mins_string = implode(NULL,$minstay[$jahr][$i]);
42     $vak_jahr = $vak_jahr.$vak_string;
43     $ueb_jahr = $ueb_jahr.$ueb_string;
44     $mins_jahr = $mins_jahr.$mins_string;
45 }
46 ...
47 //Strings zusammenführen und kürzen
48 $vak_alle = $vak_jahr.$vak_jahr2;
49 $ueb_alle = $ueb_jahr.$ueb_jahr2;
50 $mins_alle = $mins_jahr.$mins_jahr2;
51 $vak_alle = substr($vak_alle , 0, 365);
52 $ueb_alle = substr($ueb_alle , 0, 365);
53 $mins_alle = substr($mins_alle , 0, 365);
54 ...
55 }
```

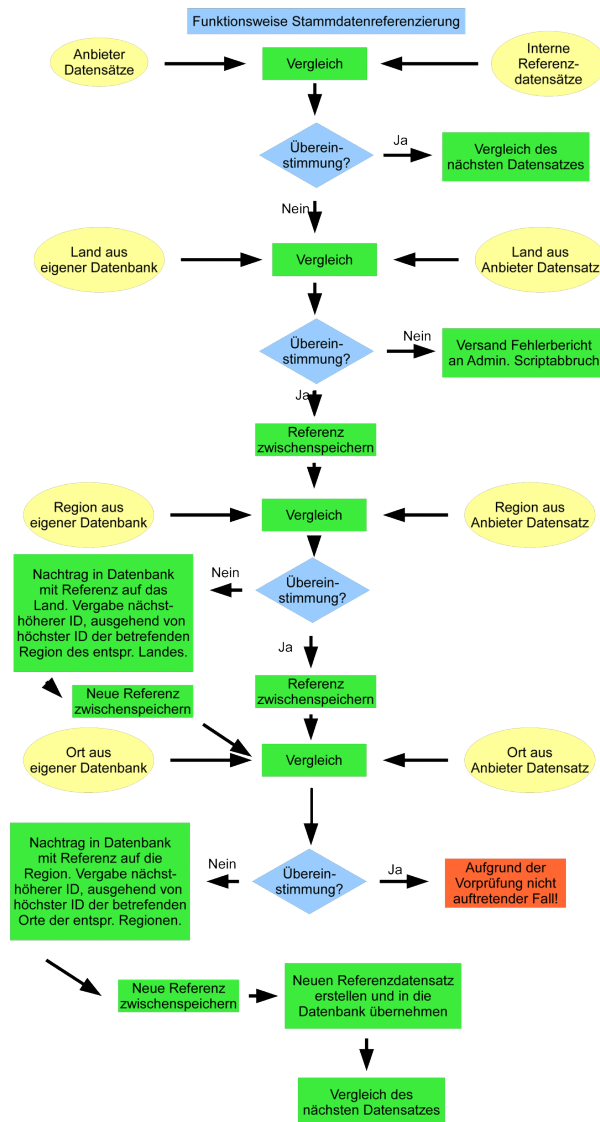


Abbildung 8.3.: Ablauf Stammdatenkonvertierung

9. Bedienoberfläche

9.1. Globale Definitionen

9.1.1. Arbeit mit einer Session

Das Buchungssystem muss in der Lage sein alle getätigten Angaben des Users solange zu erhalten, bis eine Buchung beendet bzw. eine neue Suche initiiert wird. Diese Fähigkeiten bietet das HTTP Protokoll nicht, da es sich um ein verbindungsloses Protokoll handelt.⁹⁶ Ist eine Seite komplett an den Clienten übertragen wird der Prozess geschlossen und alle zwischengespeicherten Daten gehen verloren. PHP bietet an dieser Stelle die Möglichkeit des Session-Managements⁹⁷. Jede neu gestartete Session erhält eine Session-ID, um sie eindeutig zu identifizieren. Die Zwischenspeicherung aller in der Session enthaltenen Daten erfolgt auf dem Webserver. Die Session-ID ist auf Clientseite in einem Cookie⁹⁸ abgelegt. Da nicht immer gegeben ist, dass auf Clientseite Cookies akzeptiert werden, folgt eine Übergabe zusätzlich über die HTTP-Methode GET⁹⁹. Ruft ein User die Suchmaske auf, kommt es zur Initialisierung einer neuen Session. Alle nun getätigten Eingaben werden nach dem Start der Suche an die folgende Seite übertragen. Sie nimmt die Session wieder auf, überprüft alle übergebenen Daten und schreibt sie in die Session. Bei der Folgeseite handelt es sich um die Objektübersicht. Ein Schalter im Script der Suchmaske sorgt dafür, dass auf der Folgeseite entweder Daten in der Session abgelegt oder ausgelesen werden. Ein Aufruf der Objektübersicht sollte dadurch von allen weiteren Seiten möglich sein und die zuletzt angezeigten Objekte ausgeben. Ähnlich ist das Verfahren bei Buchung eines Objektes. Alle Angaben die der User tätigt werden in die Session geschrieben.

Da es nötig ist eine Vielzahl an Daten zwischenzuspeichern, wurde die Session strukturiert. Sie beinhaltet sechs Kategorien und teilt sich in „allgemein“, „navigation“, „person“, „objekt“, „payment“ und „zusatz“ auf. Jede dieser Kategorien stellt ein Array dar. In der Kategorie „allgemein“ sind beispielsweise Angaben über Suchparameter hinterlegt. So erhält u.a. die Seite mit der Objektübersicht Informationen über vom User gewählte Ausstattungsmerkmale, Anreisedatum und gewähltes Reiseziel.

Die Kategorie „person“ übernimmt die Zwischenspeicherung von personenbezogenen Daten wie Name, Anschrift und Kontaktdaten. Unter der Kategorie „zusatz“ sind alle während der Buchung pflichtmäßig und vom User gewählten Zusatzleistungen abgelegt.

⁹⁶ Jamsa 1996, 87ff

⁹⁷ Kannengiesser 2004, 304f

⁹⁸ Kannengiesser 2004, 294f

⁹⁹ Jamsa 1996, 104f

Ihre Präsenz ist sowohl zur Preisberechnung, in der Buchungsübersicht als auch bei der eigentlichen Datenübergabe an den Anbieter notwendig.

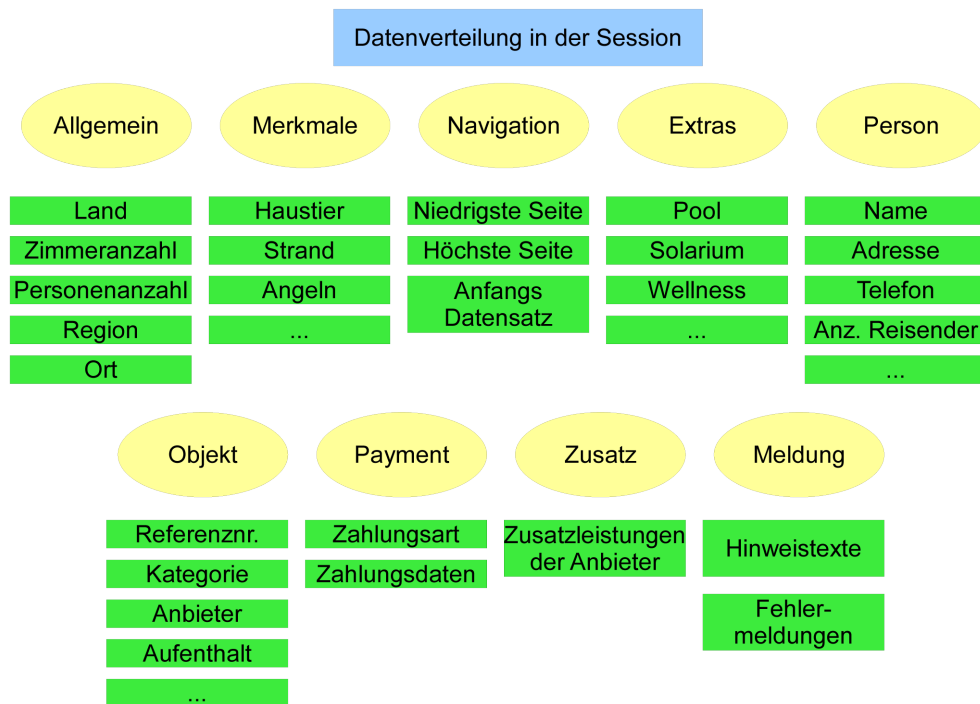


Abbildung 9.1.: Session, Datenverteilung

9.1.2. Überprüfung von Eingaben

Um Angriffe auf das System zu vermeiden werden alle vom User getätigten Angaben überprüft. Der Ablauf ist dabei immer der gleiche. Schon auf der aktuellen Seite findet mit Hilfe von Javascript eine Kontrolle statt. Erst wenn alle Eingabefehler behoben und wichtige Hinweise bestätigt wurden, ist ein Absenden der Daten mit einer der HTTP-Methoden möglich. Auf fehlerhafte Angaben folgt ein Hinweis über dem entsprechenden Feld und eine Fokussierung auf das zuletzt als fehlerhaft markierte Feld. Der Vorteil besteht darin, dass die Seite für die Überprüfung der Angaben nicht neu geladen werden muss. Andererseits könnte dieses System durch Deaktivierung von Javascript umgangen werden. Solch eine Möglichkeit bietet sich z.B. bei dem Ferienhaus Anbieter Interhome an¹⁰⁰ und stellt damit eine Sicherheitslücke dar. Hier schafft PHP Abhilfe. Nachdem alle

¹⁰⁰ Über das Web-Developer Plugin von Mozilla Firefox lassen sich diverse Analysen durchführen und bestimmte Technologien einer Web-Seite studieren und deaktivieren. Durch Deaktivierung der Interpretation von Javascript findet in der Interhome Buchungsmaske keine Kontrolle der Daten mehr

Angaben über die HTTP-Methoden GET oder POST an das Folgedokument übermittelt wurden, fängt sie ein dort inkludiertes Überprüfungsscript ab. Da es z.B. mehrere Schritte in der Buchung gibt, ist das Script in verschiedene Bereiche aufgeteilt. Der zum jeweiligen Formular passende Überprüfungsbereich wird mit Hilfe eines Parameters aufgerufen. Die Kontrolle fällt je nach Typ der Eingabe anders aus. Zeichenketten dürfen keine ungültigen Zeichen enthalten. Ungültige Zeichen sind beispielsweise spitze Klammern die zur Definition von Sprachelementen verwendet werden. In vielen Fällen darf eine Eingabe die Grenze von 50 Zeichen nicht überschreiten. Liegen längere Zeichenketten vor, wird die Eingabe auf 50 Zeichen reduziert. Diese Kontrolle findet auch in der Gegenrichtung statt. Alle Eingaben, die eine Mindestangabe von 2 Zeichen unterschreiten, zählen ebenfalls als fehlerhafte Eingabe. Eine E-Mail Adresse wird auf Existenz des Trennzeichens, dem sog. Klammeraffen, zwischen Adress- und Domainname¹⁰¹ sowie Angabe einer durch Punkt getrennten Domain überprüft.

Zahlenwerte können durch die bei vielen Angaben vorangestellten Nullen nur als String behandelt werden. Der Trick liegt darin, trotzdem alle nicht numerischen Angaben zu erkennen. Die Eingabe wird in einen Integer Wert gewandelt. Üblicherweise fallen dabei führende Nullen weg. Telefonnummern beginnen in der Regel mit einer führenden Null. Auf diese Tatsache wird bei der Überprüfung Rücksicht genommen und der Integer Wert um die fehlende Stelle ergänzt.

Weist ein Wert keine Mängel auf, landet er in der aktuellen Session. Im Falle eines Mangels ist eine Belegung der entsprechenden Session mit dem Nullwert vorgesehen. Fehlerhafte Angaben führen dazu, dass in einer Variable namens „fehler“ Meldungen abgelegt werden. Am Ende der Prüfungsschritte folgt die Auswertung der „fehler“ Variable. Ist sie mit Werten belegt, leitet das Script mittels HTTP-Header¹⁰² auf die vorherige Seite zurück. Hier folgt die Ausgabe aller Fehlermeldungen. Dieser Vorgang wiederholt sich so lange, bis alle Angaben korrekt sind. Erst dann kommt es nicht mehr zum Aufruf der HTTP-Header Anweisung und ermöglicht deshalb den weiteren Aufruf der Folgeseite.

Alle nachfolgenden Anweisungen, die mit den Werten aus dem Formular arbeiten, nehmen diese aus der Session. Dadurch ist sichergestellt, dass keine fehlerhaften Angaben für die weitere Verarbeitung zur Nutzung kommen.

9.1.3. Ansatz zur Objekt orientierten Programmierung

Da sich viele Aufgaben wiederholen, hat es sich als zweckdienlich erwiesen, einen Ansatz zur objektorientierten Programmierung zu verfolgen¹⁰³. So ist es möglich, Ressourcen und Aufwand in einem verträglichen Rahmen zu halten. Im Bereich der Datenerfassung wiederholen sich beispielsweise Vorgänge, wie das Herunterladen der Daten und die Eintragung konvertierter Daten in die Datenbank. Innerhalb einer Klasse wurden

statt. Da ein Aufruf der Folgeseiten jedoch trotzdem möglich ist liegt die Vermutung nahe dass es keine serverseitige Überprüfung gibt.

¹⁰¹ Wikipedia Domain, Stand: 3. Juni 2009, aufgerufen am 5. Juni 2009

¹⁰² Jamsa 1996, 106ff

¹⁰³ Sklar 2005, 155f

Funktionen für diese Aufgabenbereiche entwickelt. Am Beginn der Entwicklung dieser Applikation war die Benutzung einer objektorientierten Umgebung noch kein Thema. Als sich jedoch der Quellcode vervielfachte und sich eine ganze Reihe an Aufrufen wiederholte, kam nach Studium des Themas, die objektorientierte Programmierung in Betracht. Dadurch lies sich der Aufwand und Umfang der Quelltexte spürbar reduzieren und die Übersichtlichkeit erhöhen.

9.2. Suchmaske

Die Suchmaske unterscheidet in zwei Darstellungsformen. Es gibt die Standardsuche und eine Profisuche. Bei Auswahl der Standardsuche gibt es Suchoptionen wie z.B: Suche nach Land, Region und Ort. Die Aufenthaltsdauer und Reisebeginn können gewählt werden. Sucht der User nach einem bestimmten Ziel, besteht die Möglichkeit über die direkte Suche den bereits bekannten Ort, Region oder Land einzugeben. Während der Eingabe erfolgt bereits im Hintergrund eine Suche in der Datenbank. Als Entlastung wird erst gesucht, wenn der User 500ms lang keine Taste mehr drückt. Zur zusätzlichen Absicherung beginnt eine Suche erst, wenn mindestens zwei Zeichen getippt wurden. Ohne diese Sicherheit wäre die Datenbank schnell mit Anfragen überlastet. Die Live Suche beruht auf AJAX. Für die korrekte Nutzung ist eine Verarbeitung von Javascript nötig. Innerhalb der Standardsuche gibt es zusätzlich die Möglichkeit, direkt nach einem bekannten Objektcode zu suchen oder die Auswahl auf „Ferienhäuser von Privat“ zu begrenzen. Für die Bestimmung des Anreisedatums steht ein auf Javascript basierender Kalender zur Verfügung. Aus ihm ist das Anreisedatum bequem wählbar.

Die Profisuche eröffnet dem User die Möglichkeit zusätzlich aus einer Vielzahl von Optionen zu wählen. Er kann z.B. die Suche über Angaben, wie Mindestanzahl von Personen, die die Herberge fassen kann, eingrenzen. Ebenfalls ist die Mindestanzahl an Zimmern und Schlafräumen bestimmbar. Acht Ausstattungsmerkmale stehen für die Suche bereit. Optionen wie Haustier, Pool, Kamin oder Rollstuhlgeeignet können ausgewählt werden. Daneben eröffnet sich die Möglichkeit, die Entfernung die ein Objekt maximal zum nächsten Skilift, Angelplatz, Bademöglichkeit oder dem Meer haben soll, einzugrenzen. Dazu kommen neun Extras wie - Sauna, Whirlpool oder Badelandschaft.

Welcher der beiden Suchtypen zur Auswahl steht, wird durch einen Parameter bestimmt, dessen Übergabe mit Hilfe der HTTP-Methode GET stattfindet. Trägt er als Wert die „1“ erfolgt der Aufruf der Profisuche. Alle anderen Werte lassen die Standardsuche erscheinen.

Alle zur Auswahl stehenden Länder entstammen einer Abfrage aus der Datenbank. Ebenso verhält es sich mit Regionen und Orten. Ihre Ausgabe ist jedoch an die Auswahl des Landes geknüpft. Um eine Auswahl „on the fly“ zu ermöglichen, bedient sich die Seite der als AJAX definierten Elemente. Über den in Javascript definierten Event Handler¹⁰⁴ „onChange“ kommt es zum Aufruf der Funktion „UpdateDiv“, die im Javascript Teil des Scripts definiert ist. Sie erhält die zum Land gehörige ID und leitet sie mittels der HTTP-Methode GET an eine externe Datei weiter. Die externe Datei löst eine Abfrage

¹⁰⁴ Skop 2007, JavascriptDOM - JavaScript-Sprachelemente - Event-Handler

The image shows a web form for advanced search. At the top, there are two tabs: 'Standardsuche' and 'Profisuche'. The 'Profisuche' tab is active. The form is titled 'Suche'. It contains several sections:

- Suche:** Fields for 'Land:' (Land wählen), 'Region:' (Bitte zuerst ein Land auswählen), 'Ort:' (Bitte zuerst ein Land auswählen), 'Von:' (with a calendar icon), and 'Dauer' (7 Tage). There is also a checkbox for 'Nur Ferienhaus Angebote von Privat'.
- Direkt nach einem Zielort oder Objekt suchen:** Fields for 'Wunschziel:' (Wunschziel) and 'Objektcode:'.
- Räumlichkeiten:** Dropdown menus for 'Personen:', 'Zimmer:', and 'Schlafräume:', all set to 'Egal'.
- Ausstattung:** A grid of checkboxes for various amenities: Haustier, Rollstuhlgeeignet, Angeln, Bademöglichkeit, Waschmaschine, Geschirrspüler, TV, Pool, Meer, Skilifte, Golfen, Kamin. The 'Angeln', 'Bademöglichkeit', 'Meer', and 'Skilifte' options have dropdown menus set to 'Egal'.
- Extras:** A grid of checkboxes for additional features: Sauna, Solarium, Wellness, Whirlpool, Schwimmbad, Badelandschaft, Boot, Garage, Kinderbetreuung.

A 'suchen' button is located at the bottom left of the form.

Abbildung 9.2.: Suchmaske, Profisuche

aller zur Landes ID gehörigen Regionen in der Datenbank aus. Die Übertragung zwischen Script und der externen Datei lässt sich durch XMLHttpRequest¹⁰⁵ realisieren. Es ermöglicht den Transfer beliebiger Daten über das HTTP Protokoll. Die von Formularen bekannten Methoden GET und POST stehen u.a. dabei als Übertragungsform zur Verfügung. Die Rückgabe erfolgt als Objekt. In diesem Fall handelt es sich um ein HTML konformes Auswahlmenü, in welchem alle Ergebnisse der Datenbankabfrage als Auswahl zur Verfügung stehen. Die Funktion „UpdateDiv“ ersetzt im letzten Schritt das alte gegen das neue Menü. Die Auswahlfelder sind durch sog. Div Container eingeschlossen, welche eine ID besitzen. Anhand der ID ist eine Identifizierung und ein Austausch des Inhalts der Container möglich. Der Austausch von Inhalten geschieht über die Javascript Funktion „innerHTML“¹⁰⁶. Sie bietet die Möglichkeit ganze HTML Elemente dynamisch zu verändern. Die Auswahl einer Region setzt den gleichen Prozess in Gang und erneuert das Auswahlmenü für die Orte.

Wie bereits erwähnt bietet Ferienhaus.de eigene Ferienobjekte, sog. Häuser von privat. Eine Einschränkung der Suche auf diese Häuser ist möglich. Wird das entsprechende

¹⁰⁵ Holzner 2006, 29 u. 97ff

¹⁰⁶ Skop 2007, JavascriptDOM - Objektreferenz - all - innerHTML

Feld gewählt, folgt die Übermittlung eines Parameters, der die Suche auf Objekte deren Referenzcode mit 'ONFH' beginnt beschränkt.

Dem User wird die Möglichkeit geboten direkt nach einem Land, einer Region oder Ort zu suchen. Dies ist besonders praktisch, wenn das Reiseziel bereits bekannt ist. Das Textfeld ist wie auch die Auswahlmenüs an Event Handler geknüpft und bekommt Rückgabewerte auf Eingaben über AJAX.

Definiert sind zwei Event Handler. Der erste, „onClick“, sorgt dafür, dass im Feld vorhandener Text gelöscht wird. Der zweite Event Handler ruft die Funktion „liveSearch-Start“ auf, wenn eine Taste gedrückt wird. Als Übergabewert dient der Wert der Taste die gedrückt wird. Innerhalb dieser Funktion erfolgt ein Aufruf von „UpdateDiv3“. Als Input dient „UpdateDiv3“ der Wert welcher gerade im Textfeld steht. Der Aufruf dieser Funktion erfolgt mit einer Verzögerung von 500ms. Der Zähler wird bei jedem Tastendruck auf Null zurück gesetzt und startet von neuem. Hintergrund dieser Verzögerung ist die Entlastung der Datenbank von unnötig vielen Anfragen. Zusätzlich muss eine Eingabe mindestens zwei Zeichen enthalten, um eine Abfrage auszulösen. Innerhalb der Funktion kommt es zum Aufruf einer externen PHP Datei. Sie erhält über die HTTP-Methode GET die nötigen Parameter und liefert Ergebnisse als Auswahlmenü zurück, d.h. auch hier findet eine asynchrone Datenübertragung statt.

Als letzte Möglichkeit bietet sich die Suche über den Objektcode an. Hat sich der User bereits einige Häuser ausgesucht, ist es über diese Suchfunktion möglich, direkt das gewünschte Objekt aufzurufen.

Die Auswahlboxen und Entfernungsangaben sind statisch. Einige Angaben sind versteckt und definieren Werte für die Navigation auf der Folgeseite. Über weitere Parameter wird das Verhalten der Folgeseite gesteuert und die Session ID übermittelt. Der User gelangt nach dem Absenden der Suchparameter auf die Seite mit der Objektübersicht.

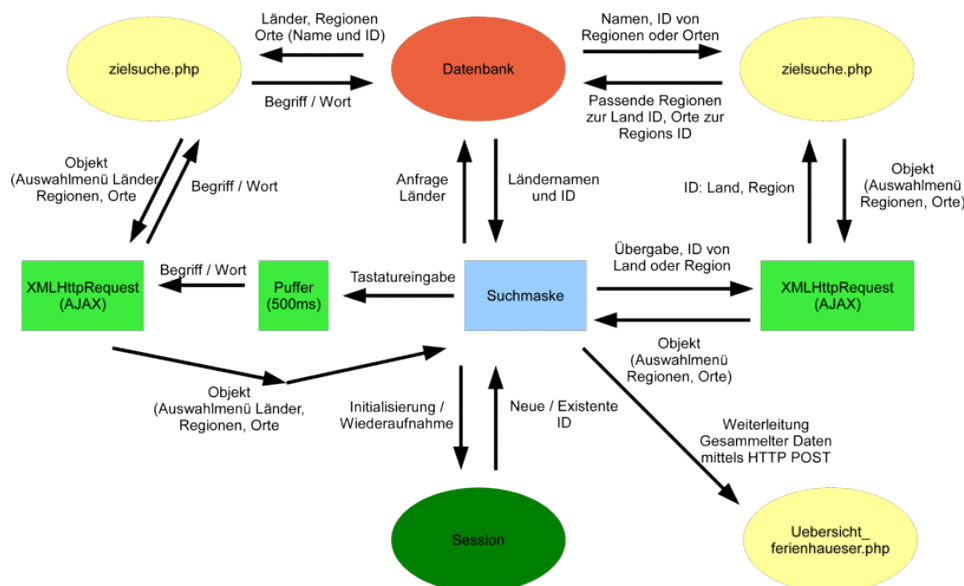


Abbildung 9.3.: Programmablauf Suchmaske

9.3. Objektübersicht

Über der Seite befindet sich die Hauptnavigation. Sie ermöglicht ein Navigieren innerhalb des Buchungssystems. Die angezeigten Menüpunkte variieren dabei auf den einzelnen Seitentypen. Die eigentliche Seite beginnt mit der Seitennavigation. Pro Seite lassen sich 10 - 25 Objekte darstellen. Standardmäßig können max. 10 Seiten über die Navigation angewählt werden. Je nach Anzahl verfügbarer Objekte und ausgewählter Seite wandert der Auswahlbereich mit. Hier ein Beispiel.: Bei einer Ausgabe von 10 Objekten pro Seite werden für 200 Objekte 20 Seiten benötigt. Zu Beginn sind die Seiten 1 bis 10 auswählbar. Die Auswahl von Seite 10 beispielsweise führt dazu, dass nun alle Seiten im Bereich von 5 - 14 wählbar sind. Ein Aufruf von Seite Nr. 5 rückt nun wieder die Seiten von 1 - 10 in den Fokus.




Suche		Merkliste	
Zurück		Vor	
Anzahl Suchergebnisse		10	
Chalkidiki, Pefkochori Nur 3 km vom beliebten Urlaubsort Pefkochori mit Tavernen, Geschäften und zahlreichen Unterhaltungsmöglichkeiten... mehr	Endreinigung 4 Personen 1 Schlafraum 350m zum Strand 350m bis zum Meer Flexibel buchbar Kurzaufenthalte möglich	ab 10.02.2009 7 Tage Preis: k.A.	 SKG01001.CYYYYP
Chalkidiki, Pefkochori Nur 3 km vom beliebten Urlaubsort Pefkochori mit Tavernen, Geschäften und zahlreichen Unterhaltungsmöglichkeiten... mehr	Endreinigung 3 Personen 350m zum Strand 350m bis zum Meer Flexibel buchbar Kurzaufenthalte möglich	ab 10.02.2009 7 Tage Preis: k.A.	 SKG01001.UYYYYO
Chalkidiki, Fourka Im beliebten Urlaubsort Fourka, an der Westküste der Halbinsel Kassandra. Nur 20 m vom feinen Sandstrand entfernt... mehr	Endreinigung 4 Personen 1 Schlafraum 20m zum Strand 20m bis zum Meer Flexibel buchbar Kurzaufenthalte möglich	ab 10.02.2009 7 Tage Preis: k.A.	 SKG01002.CYYYYO

Abbildung 9.4.: Objektübersicht

Die Ausgabe für jedes einzelne Objekt gliedert sich in vier Bereiche. Angefangen von links gibt es in der 1. Spalte Informationen zur Lage und eine Beschreibung über das Objekt. Die zweite Spalte informiert über Merkmale von Objekt und Umgebung und gibt bei einigen Objekten zusätzlich eine Bewertung aus. Soweit vorhanden liegen in der dritten Spalte Preisangaben für den in der Suchmaske genannten Urlaubszeitraum vor. In der letzten Spalte ist ein Vorschaubild und die Objektnummer zu sehen. Mit Hilfe einer Schleife lassen sich 10 oder mehr Objekte ausgeben, ohne den Programmieraufwand in die Höhe zu treiben.

Bevor die Ausgabe beginnt, steht eine Kontrolle der Sucheinstellungen an. Für diesen Schritt ist die externe Datei „varcheck.php“ inkludiert. Sie überprüft im ersten Schritt welche Aktion vor Aufruf der Seite stattgefunden hat. Kommt der User über die Suchmaske auf die Seite, steht eine Kontrolle aller gelieferten Variablen an. Fehlerfreie Angaben gelangen in die Session, fehlerbehaftete nicht. Treten Fehler auf, wird auf die Suchmaske zurück geleitet.

Dabei spielt es auch eine Rolle, welche Art der Suche stattfand. Jede Art besitzt eine andere Priorität bei der Auswertung. An erster Stelle steht die direkte Suche über den Objektcode, gefolgt von der direkten Suche nach einem bestimmten Ziel. An letzter Stelle liegt die Suche durch Auswahl des Ziels per Menü. Diese Einteilung spielt für den weiteren Ablauf im Script eine wesentliche Rolle.

Für den Fall, dass ein bestimmtes Objekt gesucht wird, entfällt die Auswertung aller anderen Werte. Stattdessen findet eine Suche nach dem passenden Objekt statt. Bleibt die Suche ergebnislos, folgt eine Umleitung auf die Suchmaske mit der Ausgabe einer entsprechenden Fehlermeldung. Im Erfolgsfall leitet das Script, unter zusätzlicher Angabe von Anbieterkennung, Objektcode und Kategorieangabe auf die Detailansicht um.

Liegt kein Objektcode als Suchkriterium vor ist die Auswertung der direkten Zielangabe an der Reihe. Als Voraussetzung gilt, dass ein Suchbegriff mindestens 3 Zeichen lang ist. Aus dem Begriff ergibt sich der Referenzcode. Die Auswertung des Codes liefert Rückschlüsse, um welche Ebene es sich handelt. Alle Codes, die bis 3 Zeichen lang sind beschreiben Länder, mit 4 Zeichen Länge Regionen und alle siebenstelligen Codes stellen Orte dar. An den Code sind weitere Anfragen geknüpft. Hat der User ein Land gewählt, werden alle Objekte des Landes angezeigt. Liegt als Anfrage eine Region oder gar ein Ort vor, sind ID des passenden Landes bzw. auch noch der passenden Region zu ermitteln und in der Session abzulegen. Nur dann ist eine Objektsuche durchführbar. Diese folgt im Anschluss und listet alle Objekte auf, die den Suchkriterien entsprechen.

Bleiben die Formularfelder für direkte Objektsuche und gezielte Suche unbenutzt, läuft die Verarbeitung auf die dritte Variante hinaus. Die Angaben von Land, Region und Ort über das Auswahlmenü.

Nach Abschluss der Auswertung, vorausgesetzt das Script für den Variablencheck leitet nicht auf eine andere Seite um, beginnt die eigentliche Ausgabe.

Alle für die Ausgabe wichtigen Angaben in der Session werden in Variablen kopiert und sind während der Laufzeit im Script verfügbar. Dieser Schritt erweist sich als vorteilhaft. Da im weiteren Verlauf sowohl auf direkten Zugriff der Daten in der Session verzichtet werden kann, als auch bei einer möglichen Änderung der Sessionvariablen nur die anfängliche Zuweisung im Script geändert werden muss.

Um Objekte auszugeben ist eine Datenbank Anfrage nötig. Da sie abhängig von den gewählten Kriterien des Users ist, sieht jede Anfrage anders aus. Sie muss also im Vorfeld dynamisch erstellt werden. Die Syntax einer Anfrage ähnelt der englischen Sprachsyntax. Werte die unabhängig voneinander abgefragt werden sind mit einem „OR“ verknüpft. Ist eine Abhängigkeit gewünscht, erfolgt die Verknüpfung mit „AND“. Eine Anfrage beginnt immer mit *SELECT (items) FROM (tabelle) WHERE [...]*. Der Ansatz besteht nun darin, anhand der gewählten Einstellung die entsprechenden Kriterien zusammen mit der gewünschten Sprachsyntax nacheinander an den ersten Teil anzuhängen. Hier

gestaltet es sich als Vorteil, dass immer ein Land angegeben werden muss. Die Angabe von Verknüpfungsoperatoren¹⁰⁷ entfällt. Sie dürfen im ersten Teil auch gar nicht vorkommen, da es noch keine weiteren Kriterien gibt mit denen verknüpft werden kann. Eine Angabe von Region und Ort ist nicht zwangsläufig nötig. Mehrere „IF“ Bedingungen prüfen diese Angaben auf Existenz. Sollten Region oder Ort als Auswahl vorhanden sein, wird eine Variable mit der passenden SQL Anweisung und der ID belegt. Für den Fall dass keine Angabe gemacht wurde, liegt auf der Variable nur ein Leerzeichen vor. Da sich die Suche auf Objekte konzentrieren soll für die alle gewählten Kriterien zutreffen, sind die einzelnen Teile der Anfrage immer mit „AND“ verknüpft. Der Teil für die Abfrage eines Ortes sieht z.B. folgendermaßen aus.: *AND ort=(betreffende ortsid)*. Weitere Verknüpfungen mit „AND“ können angehängt werden. Gleiches geschieht mit allen Merkmalen, die in der Suchmaske zur Auswahl stehen. Für sie existiert eine neu entwickelte Funktion, die eine Aufbereitung als SQL konforme Anfrage durchführt.

Je nach Angaben variiert die Zeichenlänge der SQL Anweisung. Solange der User auf der Suchmaske keine Merkmale gewählt hat, unterschreitet die Anweisung eine gewisse Länge. Anhand der Zeichenlänge lässt sich darüber urteilen welche Abfrage stattfinden soll. Denn es gibt die Möglichkeit, Stammdaten und Merkmale oder nur die Stammdaten abzufragen. Eine Abfrage ohne Merkmale kann schneller durchgeführt werden.

Ob von der Möglichkeit nur Angebote von Privat einzusehen Gebrauch gemacht wird, wertet das Script anschließend aus. Sollte dies der Fall sein wird die Suche auf Objekte beschränkt, die als Anfangsbuchstaben im Objektcode 'ONFH' enthalten. Für die eigentliche Ausgabe werden zwei Anfragen an die Datenbank gestellt. Die erste Anfrage liefert alle Ergebnisse, die den Kriterien entsprechen zurück. Aus deren Anzahl lässt sich die Gesamtseitenzahl in Abhängigkeit der gezeigten Objekte pro Seite berechnen. In der zweiten Anfrage wird nur eine bestimmte Anzahl an Datensätzen beginnend ab einer definierten Position ausgelesen. Das Ergebnis dieser Anfrage enthält alle Objekte der gewählten Seite in der Objektübersicht.

Sind die Abfragen beendet beginnt der eigentliche Seitenaufbau. Die Navigation um Angebote der Folgeseiten aufrufen zu können, lässt sich durch einige Rechenschritte und eine FOR Schleife realisieren. Vorausgesetzt wird, dass die Seitennavigation dynamisch gestaltet ist. Existieren 20 Seiten und klickt ein User die momentan höchste Seitenzahl 10 an, muss auf der Folgeseite eine Auswahl höherer Seiten möglich sein. Andererseits darf sich die Navigation nicht mehr nach oben oder unten ändern, wenn die höchste bzw. niedrigste Seite ausgewählt wurde. Die Berechnung der anzuzeigenden Seiten und Behandlung von Seitengrenzen laufen folgendermaßen ab.

Im ersten Schritt folgt die Errechnung der Seitenzahl. Sie ergibt sich aus der Anzahl aller Einträge durch die angezeigten Einträge pro Seite. Das Ergebnis wird immer auf ein Ganzes aufgerundet.

$$\text{Anzahl Seiten} = \left\lceil \frac{\text{Anzahl Einträge}}{\text{Anzahl Einträge pro Seite}} \right\rceil \quad (9.1)$$

Das Aufrunden ist nötig, da eine letzte Seite, welche weniger als 10 Einträge hat, sonst

¹⁰⁷ Riccardi 2001, 206ff

durch die Berechnung wegfallen würde.

Eine weitere Variable legt fest ab welchem Eintrag die Ergebnisse angezeigt werden sollen. Wenn beispielsweise der letzte Eintrag auf der vorherigen Seite Eintrag 39 war, soll auf der aktuellen Seite eine Ausgabe ab Eintrag 40 stattfinden. Dieser Wert lässt sich aus dem Produkt einer beliebigen Seite und Einträgen pro Seite mit einer Differenz von 10 Einträgen ermitteln.

$$\text{Anfangseintrag} = \text{Beliebige Seite} * \text{Anz. Einträge pro Seite} - 10 \quad (9.2)$$

Wie bereits im Vorfeld beschrieben passt sich die Seitennavigation dynamisch an. Dabei müssen drei Kriterien berücksichtigt werden. Bei Auswahl einer hohen oder niedrigen Seitenzahl in der Navigation ist diese auf der Folgeseite als Mittel in der Navigation zu platzieren. Existieren keine weiteren Seiten nach oben oder unten soll die Seitennavigation stehen bleiben. Für diese Fälle existieren entsprechende Fallunterscheidungen. Für den Fall, dass es nicht mehr als 10 Seiten gibt, wird als Standardwert für die erste Seite in der Seitennavigation immer die Seite 1 gesetzt. Liefert die Suche so viele Objekte, dass mehr als 10 Seiten benötigt werden, gelten folgende Bedingungen.: Im Standardfall schiebt sich die Linkanzeige um drei Felder zurück, ausgehend von der angewählten Seite. Ein Aufruf der Seiten 1 bis 3 legt als Startwert Seite 1 fest und lässt damit die Navigation auf den ersten 10 Seiten stehen. Ab der 5. Seite vor der letzten Seite schiebt sich die Linkanzeige um 9 Felder zurück.

Ist die Seitennavigation aufgebaut, folgt im Anschluss eine Ausgabe der Ferienobjekte. Die Ausgabe wird mit Hilfe des SQL Befehls *LIMIT*¹⁰⁸ begrenzt. Dieser Befehl kann sowohl die Anzahl der auszugebenden Ergebnisse beschränken als auch die Ausgabe einer bestimmten Anzahl an Datensätzen unterdrücken. So lässt sich ein genau definierter Umfang ausgeben, was für eine seitenweise Aufteilung der Ergebnisse wichtig ist.

Innerhalb einer Schleife folgt der Abruf von Beschreibungstexten, Merkmalen, Bildern, Preisen und geografischen Daten der Objekte. Da die Detailansicht später auf selbige Daten zurückgreift, ist dieser Teil als Funktion in einer Klasse hinterlegt. Der Abruf von Preisen richtet sich nach der Aufenthaltsdauer und dem Anreisedatum. Sollte es keine Preisangaben in der Datenbank geben folgt auch keine Preisausgabe. Eine Angabe von Anreisedatum und Aufenthaltsdauer findet immer statt. Sind sie vom User im Vorfeld nicht festgelegt worden benutzt das Script den aktuellen Tag und eine Aufenthaltsdauer von 7 Tagen. Bilder der Objekte liegen auf den Servern der Anbieter bereit. Leider ist die Existenz der Bilder unter den Links nicht immer gegeben. Ein erster Versuch die Existenz mit der PHP Funktion *fopen* zu hinterfragen schlug fehl, da zwar eine Fehlermeldung, aber keine Falsifizierung zurückgegeben wird. Als erfolgreich erwies sich die Funktion *getimagesize*¹⁰⁹. Sie kann auf externe Dateien zugreifen und gibt im Erfolgsfall ein Array mit Informationen zurück. Eine Überprüfung auf Existenz dieses Array gibt Aufschluss über die Existenz der Datei. Im Falle einer Nichtexistenz folgt die Ausgabe eines Standardbildes.

¹⁰⁸ Kannengiesser 2004, 368ff

¹⁰⁹ Achour 2009, Funktionsreferenz - Bildverarbeitung und -generierung - Image - Image-Funktionen - *getimagesize*

9.4. Objekt Detailansicht

Die Ausgabe der Detailansicht ist in einen Header, Merkmale, Bildergalerie und Objekt sowie Wohnungsbeschreibung gegliedert. Der Header gibt über die Lage des Objektes, seine Bezeichnung, Objektnummer und Vorschaubild Auskunft. Zusätzlich bietet er Optionen, um das Objekt in eine Merkliste aufzunehmen, Vakanzen abzufragen oder eine Buchung zu starten. Merkmale geben Auskunft über Räumlichkeiten, Ausstattung, wichtige Einrichtungen in der Nachbarschaft und enthalten z.T. Bewertungen. Die Bildergalerie umfasst bis zu 16 Vorschaubilder, welche anwählbar sind und als Großansicht im Header angezeigt werden. Da die Bilder vom Anbieter stammen, folgt vor der Ausgabe der Galerie eine Bearbeitung. Hierbei wurde auf korrekt angepasstes Format und Größe Wert gelegt. Für Bildoperationen steht in PHP die GD Library¹¹⁰ bereit auf welche im Verlauf der Bearbeitung zurückgegriffen wurde. Auf die Bildergalerie folgen Beschreibungen von Objekt und Wohnung.

Suche
Übersicht
Merkliste


Ferienhaus in Gerakini
Chalkidiki, Griechenland

Objekt:

SKG02003.ZVYYQ

Merken
Buchen

Vakanzen



Merkmale

Endreinigung

Geschirrspüler

Haustier

Kamin

Waschmaschine

Bewertung

Räumlichkeiten


3 Personen

3 Schlafräume

Entfernungen

250m zum Strand

250m bis zum Meer



Das Haus:

Nur 250 m vom Sandstrand entfernt, den Sie durch eine Straßenunterführung erreichen. Durch die zentrale Lage lassen sich beide Halbinseln bequem erreichen. Das bietet Ihre Anlage: Die familiär geführte Anlage mit insgesamt 15 Appartements und 3 Reihenhäusern wurde nach traditioneller Architektur auf einem 1 ha großen Gartengrundstück gebaut. Frühstücksraum, Aufenthaltsraum, Fernsehzimmer, Poolbar, Rezeption, Safe, Gemeinschaftspool, Kinderbecken, Sonnenterrasse, Liege, Sonnenschirme, Spielplatz. Wir empfehlen die Buchung eines Mietwagens. Transfer zubuchbar. So wohnen Sie. Alle Appartements und Reihenhäuser sind gepflegt eingerichtet mit Meerblick und verfügen über TV, Dusche/WC, Babybett (auf Anfrage).

Gerakini 1 km, Thessaloniki 70 km. Einkaufsmöglichkeiten und Taverne 1 km, Supermarkt 250 m.

Die Wohnung:

Wohnung: Reihenhäuser mit Kamin, Klimaanlage. Wohn-/Esszimmer (2 Schlafgelegenheiten). 3 Schlafzimmer (1 Doppelbett), (2 Betten), (2 Betten). Küche mit Kühlschrank, Elektroherd, Backofen, Kaffeemaschine, Wasserkocher. 2. Bad im Obergeschoss mit Badewanne/WC. Veranda, 3 Balkone. Zusatzangebot: Wahlweise vorab buchbar (€): Frühstück 7,- pro Person/Tag (kontinental). Rabatte: Frühbucher: 10% Ermäßigung bei Festbuchung bis zum 31.03. gültig für Anreisen vom 15.05.-25.05. 10% Ermäßigung bei Festbuchung bis zum 31.03. gültig für Anreisen vom 25.09.-15.10.

Abbildung 9.5.: Detailansicht eines Objekts

¹¹⁰ Sklar 2005, 515f

Wie auch in der Objektübersicht steht vor dem eigentlichen Aufruf des Scripts eine Überprüfung an. Ihre Existenz ist damit begründet, dass ein Aufruf der Detailansicht auch vom Suchformular aus möglich ist und damit sämtliche Überprüfungen, die im Script der Objektübersicht eingebunden sind, nicht ausgeführt werden. Das Überprüfungsscript schreibt außerdem alle nötigen Informationen wie beispielsweise Objektcode, Anbieterkennung und Aufenthaltsdauer in die Session. Zudem muss das Navigationsmenü über welches Zugriff auf die Übersicht, Suchmaske oder Merkliste möglich ist berücksichtigt werden. Folgt ein Direktaufruf darf kein Link zur Objektübersicht auftauchen. Welche Links angezeigt werden wird über einen Parameter bestimmt. Für einen direkten Sprung von der Suchmaske existiert deshalb eine andere Anweisung als für einen Aufruf über die Objektübersicht.

Um den Aufwand gering zu halten folgt auch im Script der Detailansicht eine Übernahme aller nötigen Werte aus der Session in lokale Variablen. Wie bereits in den Ausführungen zur Objektübersicht beschrieben existiert für die Ermittlung von geografischen Angaben, Bildern, Merkmalen und Texten eine Klasse mit mehreren entsprechenden Funktionen. Sind alle Daten gesammelt, erfolgt die Ausgabe. Die Existenz von Bildern wird dabei wie bereits beschrieben mit Hilfe von *getimagesize* überprüft. Im Header findet sich außerdem ein Info Feld welches Fehlermeldungen und Hinweise ausgibt. Allgemein gibt es Auskunft über den Status eines Objektes, wenn dieses in die Merkliste übernommen wird.

Da die Merkmale im Vorfeld in ein Array geschrieben wurden lassen sie sich über eine Foreach Schleife ausgeben. Sollten dabei leere Felder im Array auftreten werden diese ignoriert. Bewertungen liegen gleichfalls als Array vor sofern es auch zugehörige Eintragungen in der Datenbank gibt. Durch eine Überprüfung auf Existenz eines Arrays lässt sich die Ausgabe steuern. Ist eine Ausgabe möglich, durchläuft jedes Bewertungskriterium weitere Prüfungen. Bewertungen staffeln sich nach dem Schulnotenprinzip wobei ein hoher Wert eine gute Benotung darstellt. Sollte als Wert allerdings eine „0“ genannt werden, existiert für diesen Punkt keine Bewertung. Diese Einträge fallen bei der Prüfung heraus und werden nicht angezeigt.

Ist die Auswertung und Ausgabe abgeschlossen fährt das Script mit dem Aufbau einer Bildvorschau fort. Leider gibt es hierbei eine besondere Problematik. Teilweise wiederholen sich die Links bzw. liegen die selben Bilder unter verschiedenen Namen vor. Trotz unterschiedlicher Bezeichnung würden Bilder mehrmals in der Vorschau auftreten. Es muss ein Algorithmus zum Einsatz kommen der nach Dateiinhalt vergleicht. Nach mehreren Versuchen, die zu viele Ressourcen belegten, konnte eine elegante Lösung gefunden werden. Nach dem positiven Check auf Existenz kommt es zur Bildung eines Hashs¹¹¹ der Datei. Es folgt ein Vergleich mit einem Array in dem alle bisher ermittelten Hashs abgespeichert sind. Desweiteren ist ein Schalter installiert der die Bildausgabe steuert. Sollte der Hash mit einem Hashwert aus dem Array übereinstimmen, wird die Bildausgabe unterdrückt. Ist kein entsprechender Hash im Array vorhanden, findet die Ausgabe des Bildes statt und der Hash wird im Array ergänzt. Vor der Ausgabe durchläuft das Bild, wie bereits erwähnt, eine Anpassungsfunktion. Pro Zeile dürfen nur 4 Vorschau-

¹¹¹ Menezes 1996, 321ff

bilder angezeigt werden, danach soll ein Zeilenumbruch erfolgen. Während der Ausgabe läuft deshalb eine Zählvariable mit. Hat sie einen bestimmten Wert erreicht, erfolgt ein Zeilenumbruch und die Variable wird auf ihren Ursprungswert zurückgesetzt. Die Auswahl der Vorschaubilder und deren Anzeige im Header ist übrigens auch über AJAX gesteuert und lässt dadurch die Bildauswahl ohne neu laden der Seite zu. Ein Klick auf das Vorschaubild ruft die Funktion *UpdateDiv* auf welche im Head des Scriptes definiert ist. Sie erhält den Link zum Bild und gibt diesen an eine externe Datei weiter. Ein Objekt, welches das bearbeitete Bild (integriert im entsprechenden HTML Code) liefert, bildet die Rückgabe. Der Inhalt des Div Containers welcher das aktuell angezeigte Bild repräsentiert, wird mit dem neu gelieferten Code ersetzt.

Im letzten Schritt folgt die Ausgabe der Beschreibungstexte. Sie gestaltet sich einfach, da nur auf Existenz und Mindestlänge der Texte überprüft werden muss.

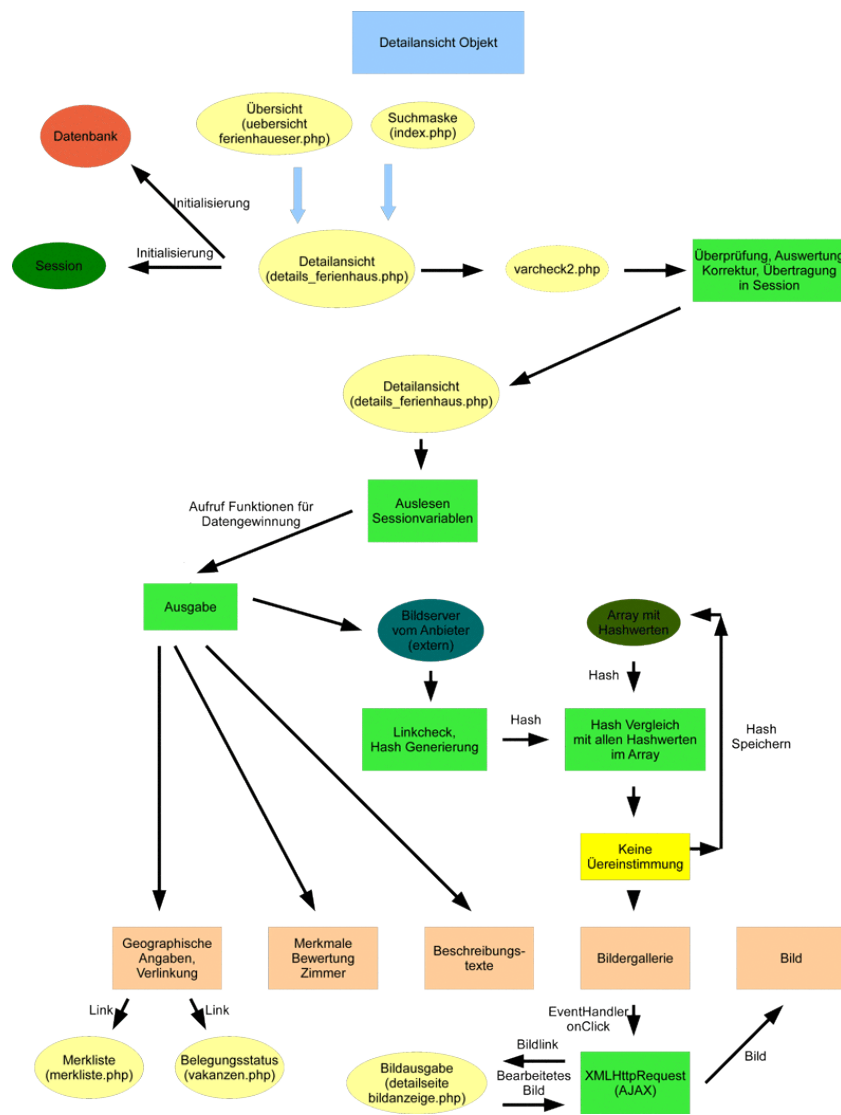





Abbildung 9.6.: Ablauf Detailansicht

9.5. Merkliste

Der optische Aufbau der Merkliste ist ähnlich dem der Objektübersicht. Die Angaben zu jedem Objekt sind auch hier in 4 Spalten aufgeteilt. Es erfolgt jedoch nur eine Ausgabe an Objekten, die durch Cookies beim User hinterlegt wurden. Die Ausgabe ist um Möglichkeiten erweitert einzelne Objekte aus der Merkliste oder die gesamte Merkliste zu löschen. Jedes Objekt ist wie in der Übersicht mit einer Detailansicht verlinkt. Maximal sind 25 Objekte auf der Merkliste möglich, mit einer Vorhaltezeit von 2 Wochen.

Suche
Übersicht

Chalkidiki, Gerakini Nur 250 m vom Sandstrand entfernt, den Sie durch eine Straßenunterführung erreichen. Durch die zentrale Lag... mehr	Endreinigung Geschirrspüler Haustier Kamin Waschmaschine 8 Personen 3 Schlafräume 250m zum Strand 250m bis zum Meer	ab 10.02.2009 7 Tage Preis: k.A.	
<input type="checkbox"/>			
Chalkidiki, Gerakini In direkter Strandlage am Ortsrand von Gerakini. Das bietet Ihre Anlage: 22 Wohneinheiten, erbaut 2001. S... mehr	Endreinigung Haustier 4 Personen 20m zum Strand 20m bis zum Meer	ab 10.02.2009 7 Tage Preis: k.A.	
<input type="checkbox"/>			
Chalkidiki, Vourvourou In herrlich bewachsener Umgebung direkt neben einem grünen Pinienwald auf Gartengrundstück (5000 qm). Hier ... mehr	Endreinigung Haustier 7 Personen 2 Schlafräume 800m zum Strand 800m bis zum Meer	ab 10.02.2009 7 Tage Preis: k.A.	
<input type="checkbox"/>			

Ausgewählte löschen
Bestätigen

Abbildung 9.7.: Merkliste

Das Script ist so aufgebaut, dass es neben der Darstellung von gespeicherten Objekten die Fähigkeit besitzt, neue Objekte einzutragen und damit neue Cookies zu setzen, bzw. einzelne oder alle Cookies zu löschen. Welche Aktion durchgeführt wird ist über einen Parameter steuerbar. Für den Fall, dass er keine oder falsche Angaben enthält, folgt immer eine Ausgabe der gespeicherten Objekte.

Die Aktionen sind als Funktion innerhalb einer Klasse im Script definiert. An die Funktionen *eintragen* und *loeschen* muss eine Zeichenkette übergeben werden, die Objektcode, Kategorie- und Anbieterkennung enthält. Als Trennzeichen zwischen den Werten dient der Unterstrich. Funktionen wie *auslesen* und *allesloeschen* benötigen dagegen keine Eingabewerte.

Die Funktion *eintragen* ruft eine Schleife mit 25 Durchgängen auf. Alle Cookies zur Merkliste sind als eine Art Array nummeriert. Jedes dieser Cookies wird auf Wert und Zeichenlänge überprüft. Liegt ein Cookie vor, welches weniger als 5 Zeichen an Code enthält oder auf 0 gesetzt ist, wird der alte mit dem neuen Wert ersetzt. Die Suche nach einem „freien“ Cookie ist mit einem Schalter verknüpft. Er verhindert, dass **alle** „freien“ Cookies mit dem neuen Wert belegt werden. Nach einer erfolgreichen Eintragung schaltet er um und verhindert weitere Belegungen. Zusätzlich wird in die aktuelle Session ein Hinweistext abgelegt. Er bestätigt bei Erfolg die Übernahme des Objektes in die Merkliste. Findet die Funktion während der Suche nach einem freien Platz in der Liste einen identischen Eintrag, wird keine Neueintragung vorgenommen. Der User bekommt in diesem Fall einen Hinweis, dass das Objekt schon in der Merkliste vorhanden ist. Anschließend leitet das Script mittels HTTP Header Anweisung auf die Detailansicht zurück.

Ein Aufruf der Funktion *auslesen* führt ebenfalls eine Schleife mit 25 Durchgängen aus. Alle Cookies die eine Zeichenlänge von mind. 5 Zeichen aufweisen, werden verarbeitet. Mit Hilfe der PHP Funktion *explode* erfolgt die Aufspaltung der Zeichenkette nach Objektnummer, Kategorie und Anbieter und eine Speicherung dieser Angaben in einem Array. Über sie lassen sich nun alle weiteren Daten für eine Ausgabe referenzieren. Vor der Referenzierung folgt eine Überprüfung der Werte auf Vollständigkeit. Sind alle Angaben in Ordnung, lassen sich mit Hilfe mehrerer Funktionen aus der *datensätze* Klasse Stammdaten, Bilder, Texte, Merkmale und Preise zu jedem Objekt ermitteln.

Die Funktion für Löschvorgänge von einzelnen Objekten vergleicht innerhalb einer Schleife die gegebene Zeichenkette mit der Kette aus den Cookies. Gibt es eine Übereinstimmung wird das betreffende Cookie auf 0 gesetzt. Bei einer Löschung der gesamten Liste sind anschließend alle Cookies auf Null.

Ist die Liste vollständig aufgebaut findet sich zu jedem Eintrag eine Checkbox. Ihre Bezeichnung leitet sich aus der Feldnummer ab. Als Wert enthält jede Box den aus drei Teilen bestehenden Code.

9.6. Belegungsübersicht, Buchungsvorbereitung

Ruft ein User die Belegungsübersicht auf, wird ihm neben den üblichen Daten im Header auch sein gewähltes Anreisedatum, Aufenthaltsdauer und Preis des Objektes für diesen Zeitraum angezeigt. Je nachdem welcher Buchungsschritt gerade ausgeführt wird, erneuern sich die Informationen oder es kommen neue hinzu. Eine Ausgabe von gewählttem Reisezeitraum und Aufenthaltsdauer findet nochmals nach dem Header auf der Seite statt, gefolgt von einem Auswahlmenü für den Anreisemonat. Die Ansicht präsentiert sich mit einem Kalender der jeweils 2 Monate ausgibt. Zum einen den aus dem Menü gewählten Monat und den Folgemonat.

Trifft der User eine Auswahl aktualisiert sich der Kalender auf den gewählten Monat. Die Aktualisierung läuft wieder über AJAX ab. Die Tage auf dem Kalender sind farblich markiert und geben Auskunft über den Belegungsstatus. Es gibt drei Farben. Eine davon signalisiert, dass eine Buchung am gezeigten Tag möglich ist, eine andere wiederum

Suche


Übersicht

Merkliste

Ferienhaus in Gjøl
Das Land am Limfjord, Dänemark
Objekt:
DK1062.8.1

Aufenthalt vom 18.07.2009
für 14 Tage
Objektpreis: 1434 EUR

Beschreibung



Ihre momentane Auswahl:
Anreisedatum: 18.07.2009
Aufenthalt: 14 Tage
Bitte Anreisemonat wählen
Februar 2009 - März 2009

Buchen

Vakanzen

Juli 2009

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

August 2009

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Objekt frei

Objekt belegt

Auf Anfrage

Zusatzleistungen:

Abbildung 9.8.: Belegungsübersicht

das keine Buchung getätigt werden kann und die dritte, dass nur eine Anfrage möglich ist. Anreisetage sind zusätzlich verlinkt und verweisen auf eine Preisübersicht, die sich mit mehreren Aufenthaltszeiträumen und zugehörigen Preisen präsentiert. Sie sind wiederum anwählbar und eine Auswahl übernimmt den präsentierten Anreisetag und den Aufenthaltszeitraum. In den meisten Fällen ist eine Anreise nur an Samstagen möglich. Zwischen Buchung und Anreise muss mindestens eine Zeitspanne von 4 Wochen liegen. Solange dieser Zeitraum unterschritten wird, ist der Verweis auf die weitere Buchung ausgeblendet. An dieser Stelle taucht ein Hinweistext auf, der den User auffordert einen späteren Anreisezeitpunkt zu wählen.

Am Beginn des Scriptes erfolgt eine Prüfung der Eingaben und gesetzten Werte. Ist beispielsweise kein Anreisedatum gesetzt, wird das aktuelle Datum als Termin eingesetzt. Die Funktion *zusatzleistungen_holen* stellt alle Zusatzleistungen für die spätere Ausgabe bereit. Über weitere Funktionen erfolgt u.a. die Konvertierung von Datenformaten. Sämtliche Funktionen in PHP, die mit Datumswerten arbeiten, benötigen ein Datum in der Form YYYY-MM-DD. Da das deutsche Format in seiner Form DD-MM-YYYY abweicht

sind Konvertierungen notwendig. Die Funktionen für die IBE sind so ausgelegt, dass sie auch den Unix Timestamp ausgeben. Dieser Zeitstempel zählt die Sekunden seit Beginn der Unix Epoche¹¹². Für PHP Anwendungen unter Windows beginnt die Zählung am 1. Januar 1970. Der Unix Timestamp eignet sich besonders gut um Datumsangaben in Rechenoperationen einzubinden, was z.B. die Möglichkeit ergibt, Zeitabstände zwischen mehreren Datumsangaben zu berechnen. Im vorliegenden Fall wird daraus die Zeitspanne zwischen Buchungstag und Anreisetag ermittelt. Sie darf einen Zeitraum von 4 Wochen nicht unterschreiten. Auf die Konvertierung folgen Bildschirmausgaben, die dem User u.a. ein Auswahlmenü mit Monatsangaben präsentieren. Die Auswahl enthält 12 Einträge, entsprechend den Monaten, dynamisch erstellt über eine Schleife und beginnend ab dem aktuellen Monat. Jeder Eintrag setzt sich aus dem aktuellen und folgenden Monat zusammen. Die Auswahl ist wie in vielen weiteren Fällen als AJAX Konstrukt über Javascript mit dem Aufruf einer externen Datei verbunden. Diese erhält als Werte den ausgewählten Monat in der Form 15.MM.YYYY, Referenz- und Kategoriecode des Objektes sowie Anbieterkennung, Aufenthaltsdauer und Session ID. Sie hat die Aufgabe aus den erhaltenen Angaben, die passende Aktion für den jeweiligen Anbieter auszuführen. Aktionen können *kalender* und *preis* sein. Die Aktion *kalender* führt unter Berücksichtigung der spezifischen Eigenschaften des Anbieters eine Vakanzabfrage durch. Auf die Aktion *preis* folgen Preisausgaben für das Objekt unter Voraussetzung eines vorher gewählten Anreisetages. Preis- und Vakanzabfragen werden aus getrennten Scripten aufgerufen. Für jeden Anbieter sind eigene Scripte definiert. Im Beispiel ergeben sich 4 Scripte für 2 Anbieter. Eine Auswahl führt zur Aktualisierung des Kalenders, der sich unterhalb der Auswahl befindet. Eine Funktionsbeschreibung des Kalenders folgt in den Unterkapiteln. Die Auswahl ist von einem Button gefolgt, der den nächsten Schritt in der Buchung ausführt. Er ist wie bereits erwähnt an die Prüfung der Zeitspanne zwischen Buchungs- und Anreisetag geknüpft. Dazu wird auf den Unix Zeitstempel des Buchungszeitpunktes ein Wert von 1814400 Sekunden addiert. Damit liegt das errechnete Datum 3 Wochen in der Zukunft. Ein Vergleich mit dem Unix Zeitstempel des Anreisetages entscheidet über den Status des Verweises der zum nächsten Schritt führt. Solange der errechnete Zeitstempel einen größeren Wert besitzt als der Anreisetag, ist der Verweis zum nächsten Schritt ausgeblendet und durch einen Hinweistext ersetzt. Nur wenn der Wert des Zeitstempels kleiner und damit der Zeitraum bis zum Anreisetag größer als 3 Wochen ist, sind weitere Buchungsschritte möglich.

9.6.1. Vakanzausgabe Interhome

Als Kommunikationsschnittstelle zu den Webservices nutzt Interhome das SOAP Protokoll. Auch dafür stellt PHP einige Funktionen bereit. Über die Funktion `__soapCall`¹¹³ lässt sich ein SOAP Aufruf durchführen. Bedingung ist, dass im Vorfeld eine Verbindung zum Webservice aufgebaut wurde. Dazu ist eine Authentifizierung bei betreffendem Service nötig. Nach dem Verbindungsaufbau wird ein Header mit Benutzername

¹¹² Wikipedia Timestamp, Stand: 15. Februar 2009, aufgerufen am 20. April 2009

¹¹³ Achour 2009, Funktionsreferenz - Web Services - SOAP - SoapClient- `__soapCall()`

und Passwort verschickt. Ist die Anmeldung erfolgreich, sind weitere Anfragen an den Webservice möglich. Andernfalls verweigert er weitere Auskünfte.

Bevor die erste Abfrage beginnt sind einige Berechnungen nötig. Das Anreisedatum wird aufgeteilt und als Array zurückgegeben. Das Array ist Voraussetzung, um getrennt auf die Einheiten Tag, Monat und Jahr zurückgreifen zu können. Die Werte Monat und Jahr werden zudem noch in eigenen Variablen erfasst. Für die Erzeugung des Kalenders muss bekannt sein, wie viele Tage der Monat hat für den der Kalender erstellt wird. Da sich auch dieser Schritt in mehreren Scripten wiederholt, wurde für die Berechnung eine Funktion namens *days_act_month* geschrieben. Sie ermittelt aus dem Array die Anzahl an Tagen des Monats. Im nächsten Schritt ist die Bestimmung des ersten und letzten Tages des Monats an der Reihe. Für den ersten Tag ist dies kein Problem, da er immer mit dem 1. beginnt. Der letzte Tag des Monats lässt sich aus der Gesamtzahl der Tage des Monats ableiten und ist somit auch bekannt. Mit Hilfe dieser beiden Werte lässt sich das Datum des ersten und letzten Tag des Monats bilden. Die Berechnung bzw. Erzeugung der beiden Daten übernimmt die Funktion *startendactmonth*. Da der Kalender sowohl aus gewähltem - als auch Folgemonat besteht, wiederholen sich die Berechnungen noch einmal. Leider muss für den folgenden Monat eine Vorprüfung auf Jahreswechsel und Datumskonformität erfolgen. Dies erledigt die Funktion *thisonextmonth* und stellt das Datum für den Folgemonat bereit. Das Anfangs- und Enddatum des Monats ermittelt die Funktion *startendnextmonth*.

Ursprünglich waren die Daten gedacht, um über die SOAP Schnittstelle eine Anfrage an Interhome zu senden und als Rückantwort die Belegungsübersicht zu erhalten. So sollte eine hohe Aktualität der Daten gewährleistet sein. Leider stellte sich heraus, dass über den Webservice nicht alle Belegungsdaten zur Verfügung stehen. Im Endeffekt bilden deshalb die Vakanzdaten eine Grundlage, welche 2x täglich aktualisiert auf dem FTP Server des Anbieters liegen. Die SOAP Abfrage wurde durch eine Datenbankabfrage ersetzt. Die Notwendigkeit der berechneten Daten bleibt jedoch bestehen. Die bereitgestellten Daten bieten nicht nur Auskunft über die Belegung, sondern enthalten auch Angaben zum Übergabestatus und der Mindestaufenthaltsdauer. Anhand des Übergabestatus lässt sich genau bestimmen zu welchen Tagen das Objekt gebucht werden kann. Die Beschränkung auf den Samstag als einzigen Anreisetag entfällt damit, erfordert jedoch eine Neukonzeption der Live Preisabfrage. Diese ist sowieso fällig, da sich eine Buchung mit den Daten über Mindestaufenthaltsdauer nicht mehr starr auf ein bis vier Wochen beschränkt. Alle drei Informationsblöcke sind als Zeichenkette kodiert und werden nach Abruf aufgeteilt und in ein Array geschrieben.

Nach der erfolgten Aufbereitung beginnt die Kalenderausgabe. Die Funktion *tabelle_monatskopf* übernimmt die Gestaltung des Tabellenkopfes und liefert den Namen des Monats und die Jahreszahl zurück. Im nächsten Schritt wird der erste anzuzeigende Tag im Kalender ermittelt. Der Kalender besteht aus 6 Zeilen a 7 Tagen und könnte damit theoretisch 42 Tage anzeigen. Da nicht jeder erste Tag im Monat ein Montag ist darf die Ausgabe nicht immer im 1. Feld beginnen. Meist bleiben auch am Ende des Kalenders einige Tage frei, da sie bereits zum kommenden Monat gehören. Um den ersten Kalendertag korrekt setzen zu können wird ein Array erzeugt dass alle Tage in englischer Schreibweise enthält. Jedem dieser Tage ist ein Wert zugewiesen, beginnend

mit 1 für Monday bis 7 für Sunday. Über die PHP Funktion *strtotime*¹¹⁴ lässt sich aus dem ersten Tag des gewählten Monats ein Unix Zeitstempel erzeugen. Eingesetzt in die Funktion *strftime*¹¹⁵, unter Angabe des Zeitstempels und dem Parameter **%A**, ergibt sich der Name des Wochentages. Mit Hilfe des Arrays lässt sich nun der zugehörige Startwert ermitteln. Aufgrund der Tatsache dass immer 42 Felder ausgegeben werden, kommt eine *For* Schleife mit 42 Durchgängen zum Einsatz. Die Ausgabe der Kalendertage ist an Bedingungen geknüpft. Erst wenn der Zähler der Schleife mindestens genauso groß ist wie der Wert für den entsprechenden Wochentag, findet eine Ausgabe statt. Ab diesem Zeitpunkt läuft ein zweiter Zähler mit, der die Tage im Monat repräsentiert. Er bildet die zweite Bedingung für eine Ausgabe. Ist er größer dem letzten Tag des Monats soll die weitere Ausgabe von Tagen wieder unterdrückt werden.

Wenn Zähler 2 \geq Tag im Array \rightarrow Ausgabe der Tage beginnen (9.3)

Solange Zähler 2 \leq Letzter Tag im Monat \rightarrow Ausgabe der Tage (9.4)

Alle Felder die außerhalb der Bedingungen liegen bleiben leer. Für die Darstellung des Kalenders kommt eine Tabelle zur Anwendung. Ihre Formatierung läuft z.T. innerhalb der Schleife ab. Dazu ist eine Bedingung gegeben, die den Zähler der Schleife überwacht. Nimmt er als Werte 8, 15, 22, 29 oder 36 an, folgt das Setzen eines **<tr>**¹¹⁶ Tags, der den Anfang einer neuen Zeile ankündigt. Beschriebene Werte stellen den Wochenanfang und damit den Beginn einer neuen Zeile dar. Eine ähnliche Bedingung prüft auf die Werte 7, 14, 21, 28, 35 und setzt entsprechend einen Tag der das Zeilenende signalisiert. Tage, an denen eine Buchung möglich ist, werden als Link ausgegeben und verweisen mit den entsprechenden Angaben auf die Preisabfrage. Der zweite Teil des Scripts funktioniert nach dem selben Prinzip, behandelt jedoch den Folgemonat.

9.6.2. Vakanzausgabe Wolters

Als Vorlage dient das Script für die Vakanzausgabe von Interhome Objekten. Anstelle der Datenbank findet eine XML Anfrage statt. Als Antwort gibt der Wolters Webservice eine XML Datei mit Belegungsübersicht zurück. Die direkte Anfrage hat den Vorteil aktueller zu sein als ihr Datenbank gestütztes Pendant, liefert jedoch keine zusätzlichen Daten zum Übergabestatus und Mindestaufenthalt zurück. Die XML Datei lässt sich über die PHP Funktionen *fopen* und *fread* in einer Variable abspeichern. Durch SimpleXML Funktionen wird daraus ein XML Objekt gebildet. Mit Hilfe einer *Foreach* Schleife lassen sich alle Vakanzen auslesen. Vakanzen liegen immer nur pro Woche vor, d.h. es existiert nicht für jeden Tag eine Aussage. Die Belegung ist immer für einen Zeitraum von 7 Tagen fest. Die Rückgabe umfasst meist mehr als den gewählten Monat. Oft sind Tage des vorherigen und folgenden Monats enthalten. Alle Vakanzdaten aus dem XML Objekt werden in einem Array erfasst. Pro Feld liegen Datum der Belegung und Status vor. Über

¹¹⁴ Achour 2009, Funktionsreferenz - Datum Uhrzeit - Funktionen - *strtotime*

¹¹⁵ Achour 2009, Funktionsreferenz - Datum Uhrzeit - Funktionen - *strftime*

¹¹⁶ Siehe Fußnote 53 Seite 34

die PHP Funktion *count*¹¹⁷ lässt sich anschließend die Anzahl an Feldern und damit die Wochenanzahl ermitteln. Eine zweite Schleife konvertiert die gelieferten Informationen in ein Format, was dem von Interhome entspricht, um damit den Rest des Scripts zu vereinfachen. Vor dem Aufruf der zweiten Schleife wird ein Schalter namens **firstrun** initialisiert. Er ist für die Aufarbeitung von Vakanzen wichtig die vor dem gewählten Monat auftreten. Die Anzahl der Aufrufe der *For* Schleife legt sich über die von Wolters gelieferte Wochenanzahl fest. Innerhalb der Schleife besteht der erste Schritt darin das erste Datum aus dem Array aufzuspalten. Eine anschließende Kontrolle des Monats gibt Aufschluss darüber, ob die Vakanzen bereits für den gewählten Monat gelten oder zu einem vorhergehenden Monat gehören. Handelt es sich um einen vorhergehenden Monat folgt der Aufruf einer weiteren Bedingung, welche den Belegungsstatus überprüft. Je nach Status bekommt ein weiterer Schalter seinen Wert zugeteilt. Ist das Objekt für den Zeitraum buchbar liegt der Schalter auf 0, andernfalls auf 1. Sollte es keinen Wert vor dem gewählten Monat geben, ist der Schalter mit der Aussage **NULL** belegt. Diese Vorprüfung ist unumgänglich, da Wolters die Belegungsübersicht nur Wochenweise ausgibt und sich ein Monatswechsel mitten in der Woche ergeben kann. Der Status liegt in diesem Fall im Vormonat und muss durch eine Hilfsvariable dem Teil im Script, der daraus eine Zeichenkette bildet, mitgeteilt werden.

Im nächsten Schritt steht die Bildung eines Arrays an, dass für jeden Tag den Belegungsstatus enthält, beginnend mit dem ersten Tag des gewählten Monats, maximal 28, 29, 30 oder 31 Zeichen lang. Im ersten Schritt werden sowohl Startwert als auch Endwert für die erste vollständig im gewählten Monat liegende Woche ermittelt. Hieraus ergibt sich, dass einige Tage vor der ersten vollständigen Woche des Monats ja eigentlich frei bleiben. Deshalb wurde eine Vorprüfung durchgeführt und der Belegungsstatus des vorherigen Monats ermittelt. Hier setzt nun eine Schleife an, die durch den Schalter **firstrun** gesteuert ist. Die Bezeichnung **firstrun** deshalb da der Aufruf nur einmal am Anfang stattfinden darf. Innerhalb der Schleife werden nun alle Felder, beginnend ab dem ersten Feld, bis zu dem Feld belegt, welches dem ersten Wochentag der ersten vollständigen Woche des gewählten Monats entspricht. Dabei muss berücksichtigt werden, dass ein Array im allgemeinen mit Feld 0 beginnt und für eine korrekte Belegung der Tag um einen Wert zu verringern ist. Der Schalter **firstrun** bekommt anschließend den Wert 0 und eine erneute Ausführung der Schleife wird verhindert. Nun folgt ein sich wiederholender Prozess, der beginnend mit dem von Wolters gelieferten Wochentag immer 7 Felder im Array belegt und mit der Auswertung der letzten von Wolters zurückgegebenen Woche endet. Im Anhang befindet sich der Quelltext des Kalenders. Er beschreibt den Aufbau und ist mit dem der Interhome Angebote identisch.

9.6.3. Vakanzausgabe Objekte von Privat

Sie funktioniert nach dem selben Prinzip wie es für Interhome beschrieben ist. Die Vakanzen liegen in der Datenbank zum Abruf bereit. Ihre vorherige Konvertierung übernimmt das Exportscript auf dem alten Ferienhaus Server.

¹¹⁷ Achour 2009, Funktionsreferenz - Variablen- und typbezogene Erweiterungen - Arrays - count

9.6.4. Preisausgabe Interhome

Das Prinzip entspricht dem der Wolters Preisausgabe. Da für Interhome jedoch inzwischen ausführliche Daten vorliegen, muss eine Anpassung an die neuen Gegebenheiten erfolgen. Berücksichtigung sollen dabei auch Informationen über Mindestaufenthalt und Übergabestatus finden. Zudem sind Anreisetage nicht mehr auf den Samstag beschränkt. Dies lässt sich auf die Schnelle nicht mehr realisieren und fließt damit in die zukünftige Entwicklung ein.

9.6.5. Wolters Preisausgabe

Für ein grundlegendes Funktionieren der Ausgabe ist das Anreisedatum wichtig. Liegt es in der deutschen Variante DD.MM.YYYY vor, folgt eine Aufsplittung mit anschließender Neusortierung. Ziel der Preisausgabe ist es, vom Anreisetag ausgehend Preise für 1 bis 4 Wochen Aufenthalt vom Anbieter abzufragen. Als Anreisetag ist bei dieser Methode immer der Samstag festgelegt. Um die 4 möglichen Abreisetage zu bestimmen wird eine Umformung des Datums in den Unix Timestamp durchgeführt. Aus dem entstehenden Wert lassen sich durch einfache Addition des entsprechenden Zeitumfangs in Sekunden weitere Timestamps bilden die dem Abreisetag nach 1 bis 4 Wochen bilden. Mit Hilfe der PHP Funktion *strtotime* lassen sich aus dem Timestamp alle Werte wie Jahr, Monat und Tag zurückgewinnen. Durch die Anordnung der einzelnen Elemente lässt sich die gewünschte Formatierung realisieren. Daten werden in der Englischen und deutschen Schreibweise zur Verfügung gestellt.

Die Abfrage findet über eine von Wolters bereitgestellte Schnittstelle¹¹⁸ in Form einer PHP Datei statt. Sie erhält als Parameter den Anreisetag und die Aufenthaltsdauer in Wochen, Code und Kategorie des Objektes, sowie die Anweisung nur Preise zu liefern. Als Antwort liefert Wolters eine XML Datei, die das Kontingent für den gewählten Zeitraum enthält. Nachdem sie erfasst und in eine Variable zwischengespeichert wurde, lässt sich aus ihr ein XML Objekt bilden und alle darin enthaltenen Daten abfragen. Da immer nur ein Preis gefragt ist wird auch nur dieser zusammen mit dem Aufenthaltszeitraum ausgegeben. An dieser Stelle sind die berechneten Daten für den Abreisetag wichtig, da Anreise und Abreisedatum angegeben sind. Angaben über den Aufenthaltszeitraum sind als Verweis ausgelegt. Folgt eine Auswahl, werden sowohl Anreisedatum als auch Aufenthaltsdauer entsprechend den Angaben in der Preisausgabe aktualisiert. Üblich sind 4 Verlinkungen, nach Aufenthaltsdauer gestaffelt. Eine weitere Bedingung sorgt dafür, dass Preise mit dem Wert 0 Euro nicht angezeigt werden. Wolters trifft zu jeder Preisangabe auch eine Aussage über den Belegungsstatus des Objektes. Dies erweist sich als praktisch da sich auf diese Art und Weise Preise ausblenden lassen, für die das Objekt schon belegt ist.

¹¹⁸ Lock 2007, 1

9.6.6. Preisausgabe Privatangebote

Eine Preisübersicht ist momentan für private Objekte nicht vorgesehen. Buchungen sind dort nur über eine direkte Anfrage möglich. Deshalb wird nur ein Hinweis angezeigt.

9.7. Buchungsmaske

Hat der User ein gültiges Anreisedatum gewählt, lässt sich die Buchung fortsetzen. Auf der nächsten Seite folgt die Erfassung von Kundendaten und Zahlungsart. Hier ist auch eine Auswahl an Zusatzleistungen möglich. Eine Bestätigung von Buchungshinweisen ist Pflicht. Ohne diesen Schritt ist ein Weiterkommen nicht möglich. Die Navigation im oberen Bereich wird durch eine Statusanzeige ersetzt und repräsentiert die Buchungsschritte. Der aktuelle Schritt ist gekennzeichnet. Der Header repräsentiert die selben Eigenschaften wie in der Belegungsübersicht.


Adressdaten weitere Optionen Angaben prüfen Buchung bestätigen	
<p>Ferienhaus in Gjel Das Land am Limfjord, Dänemark Objekt: DK1062.8.1</p>	
<p>Aufenthalt vom 18.07.2009 für 14 Tage Objektpreis: 1434 EUR</p>	
Adressdaten	
Anrede:	<input checked="" type="radio"/> Herr <input type="radio"/> Frau
Name*	<input type="text" value="Max"/>
Vorname*	<input type="text" value="Mustermann"/>
Strasse*	<input type="text" value="Musterstr 123"/>

Abbildung 9.9.: Header und erster Teil des Formulars

Im anschließenden Buchungsformular sind Pflichteingaben wie Anschrift und Telefonnummer einzutragen. Zusatzleistungen teilen sich in Leistungen auf die bezahlt werden müssen, darunter fallen z.B. Kautionen und Leistungen, welche zubuchbar sind. Einige Buchungshinweise, beispielsweise zu Kautionen, müssen bestätigt werden. Im letzten Punkt steht die Ausgabe von Preisen und Zahlungsdetails an. Meist handelt es sich dabei um den Objektpreis und einen Anzahlungspreis. Für die Zahlungsart stehen drei

Möglichkeiten bereit. Dabei handelt es sich um Zahlung mit Kreditkarte, Bankeinzug oder per Rechnung. Dazu kommen Zahlungshinweise, die je nach Anbieter variieren. Hier ist immer eine Bestätigung der AGB's des entsprechenden Anbieters Pflicht. Nur dann kann der Buchungsvorgang fortgesetzt werden. Geschäftsbedingungen lassen sich über einen Link erreichen.

Zusatzleistungen

Reise Rücktritts Kostenschutz:

☒ Kein Reise Rücktritts Kostenschutz
☐ Interhome Reise-Rücktrittskosten-Schutz 34 EUR
☐ Interhome Optimalschutz-Paket 49 EUR

Im Preis enthalten

- Endreinigung
- Haustier
- Kinderbett

Vor Ort zu bezahlen:

- Energiekosten, 0.00 EUR nach Verbrauch
- Kautionsbar, 200.00 EUR pauschal

Zusätzlich buchbar und vor Ort zu bezahlen:

- ☐ Zusatzbett, 5.00 EUR pro Person/Tag
- ☐ Toilettenwäsche, 3.00 EUR pro Person
- ☐ Bettwäsche (Erstausrüstung), 5.00 EUR pro Person

Preise und Zahlungsdetails

Objektpreis: 370 EUR für 14 Tage

Der Betrag muss bis zum 20.02.2009 beglichen werden.

Zahlungsart:

☒ Kreditkarte (Es sind Visa oder Mastercard möglich)
☐ Rechnung (Nur bis max 29 Tage vor Reiseantritt möglich)

Dieses Objekt wird bereitgestellt von Interhome.

Abbildung 9.10.: Auswahl von Leistungen und Zahlungsart

Von der Programmierung her ist das Script in zwei Bereiche aufgeteilt. Das Formular zur Erfassung von personenbezogenen Daten stellt den 1. Teil dar. Im 2. Teil folgen Ausgaben speziell für den jeweiligen Anbieter. Der Übersichtlichkeit halber sind die Abläufe in eigene Funktionen ausgegliedert und befinden sich am Anfang des Quelltextes. Je nach Anbieter kommt dann die eine oder andere Funktion zum Einsatz. Ihr Aufruf ist mit dem von Funktionen aus einer Klasse ähnlich.

Das Formular im 1. Teil enthält einige Auswahlmenüs. Hier sind exemplarisch Länder und Personenanzahl zu nennen. Alle Länder lassen sich aus der Datenbank ermitteln. Dazu existiert die Entität *menu_land*. Wie hoch die maximale Belegung mit Personen sein darf lässt sich über die Objektmerkmale bestimmen. Die eigentliche Abfrage wurde als Funktion innerhalb einer Klasse definiert und lässt sich in jedem Script in der die Klasse initialisiert wurde benutzen. Anschließend folgen Ausgaben des 2. Teils. Ihre Aufgaben und Funktionsweise ist in den folgenden Unterkapiteln beschrieben.

Einige Javascript Funktionen sorgen für die Überprüfung der Formularfelder. Sie kontrollieren die Wortlängen, Datentyp und E-Mail Adresse. Zeichen mit denen sich Code



Abbildung 9.11.: Ablauf Buchung Teil 1

ausführen lässt, werden ebenfalls ausgefiltert bzw. ein Warnhinweis ausgegeben und die weitere Verarbeitung abgebrochen. Eine zusätzliche aber wichtige Kontrolle erfolgt auf dem Server wenn das Formular abgeschickt wurde. Nur diese Kontrolle ist wirklich sicher. Bei Javascript besteht die Gefahr, dass es ein User deaktiviert und damit keine Prüfung mehr stattfindet. Die serverseitige Kontrolle legt alle Eingaben nach erfolgreichem Test in der Session ab. Fehlerhafte Eingaben führen zu einem erneuten Aufruf des Formulars unter Angabe aller Daten, die als korrekt eingestuft wurden. Das gilt im übrigen auch für Auswahlmenüs. Meist gehen sie aus Datensätzen der Datenbank hervor. Ein Vergleich mit den Angaben aus der Session legt den dort gegebenen Wert als Standardwert fest. Enthält die Session noch keine Angabe, stellt der erste erhaltene Datensatz aus der Datenbank den Standardwert dar. Über jedem Eingabefeld steht ein Platzhalter für die Fehlermeldung. Jedes Feld erhält so seine eigene Meldung, geliefert von den Javascript oder PHP Prüffunktionen. Ist Javascript aktiv, springt der Cursor auch noch in das letzte Feld, welches eine fehlerhafte Eingabe enthält.

Den Formularfeldern folgen Angaben über Zusatzleistungen, Hinweise und Zahlungsangaben. Die Ausgaben sind vom jeweiligen Anbieter abhängig und in Funktionen aufgeteilt. Je nach Anbieter kommt dann diese oder jene Funktion zum Einsatz. Der interne Ablauf der Funktionen folgt immer nach dem gleichen Schema. Erst die Zusatzleistungen, dann Zahlungsoptionen und Hinweise. Sowohl Abruf als auch Aufbereitung unter-

scheiden sich allerdings. Die Einbindung ist durch die PHP Funktion *call_user_func*¹¹⁹ möglich. Welche der Funktionen zum Einsatz kommt wird mit Hilfe der Anbieterkennung entschieden. Da eine Ausgabe und Verarbeitung in diesem Umfang scheinbar nicht an das Hauptsript zurückgegeben werden kann, findet die Ausgabe innerhalb der Funktion statt. Deshalb ist auch die Abfolge der Ausgaben nicht willkürlich anzuordnen. Über den Submit Button lassen sich die Daten versenden. Die versteckte Variable **checkup** sorgt dafür, dass im Script für die Formulkontrolle die richtigen Funktionen zum Aufruf kommen. Bei Erfolg bekommt der User ein Formular mit den Zahlungsmodalitäten zu sehen. Welche Zahlungsangaben er machen muss hängt davon ab, welche Zahlungsoption er gewählt hat.

9.7.1. Funktion „Zusatzleistungen Interhome“

Objekte, die außerhalb der Funktion zum Einsatz kommen, können nicht innerhalb der Funktion verwendet werden. Um mit ihnen zu arbeiten, ist eine weitere Initialisierung innerhalb der Funktion ein Muss. Da nicht immer gegeben ist, dass Preise die in der Datenbank enthalten sind auch noch der Realität entsprechen, folgt eine Preisabfrage über die Webservices von Interhome. Diese Preisabfrage benötigt allerdings die Aufenthaltsdauer in der Form Anreisedatum bis Abreisedatum. Da es nur ein Anreisedatum und die Aufenthaltsdauer in Tagen gibt, muss aus diesen beiden Werten ein Abreisedatum errechnet werden. Hierfür steht eine Klasse mit diversen Zeitfunktionen bereit. Erster Schritt ist die Konvertierung der Datenformate. Hierfür steht die selbst entwickelte Funktion *datum_deisoun* bereit. Sie erhält ein Datum in der Form DD.MM.YYYY und gibt es als YYYY-MM-DD und Unix Timestamp zurück. Die englische Schreibweise kann direkt als Anreisedatum in der Anfrage zum Einsatz kommen. Aus Timestamp und Aufenthaltsdauer lässt sich das Abreisedatum berechnen. Eine Konvertierung in das richtige Format übernimmt die PHP Funktion *strtotime*. Damit sind Checkin und Checkout Datum bekannt. Der Service *PriceRqst*¹²⁰ liefert zum Aufenthaltszeitraum passende Preis- und Währungsangaben zurück. Im genaueren handelt es sich dabei um festgesetzte Daten zu denen jeweils die Anzahlung bzw. eine Gesamtzahlung fällig ist. Hinzu kommen Angaben wie Preise, Arten des möglichen Reiserücktrittskostenschutzes und Währung. Alle vorliegenden Daten werden in die deutsche Schreibweise konvertiert und wie auch Preise und Versicherung in einem Array gespeichert.

Erst auf die Ausgabe des aktuellen Preises folgt die Angabe von Zusatzleistungen. Hierfür lässt sich die Funktion *zusatzleistungen_holen_getrennt* benutzen. Sie ruft nacheinander Leistungen aus den Kategorien *im Preis enthalten*, *vor Ort zu bezahlen* sowie *zusätzlich buchbar und vor Ort zu bezahlen* auf und legt sie, getrennt voneinander, in Arrays ab. Diese drei Kategorien sind so von Interhome vorgegeben¹²¹. Nun ist es an der Zeit, alle gesammelten Daten auszugeben.

¹¹⁹ Achour 2009, Funktionsreferenz - Variablen- und typbezogene Erweiterungen - Function Handling - *call_user_func*

¹²⁰ Interhome AG 2008b, 16f

¹²¹ Interhome AG 2008a, 17f

An erster Stelle steht die Versicherung. Interhome bietet zwei Arten von Schutz an. Zum einen das Standard Paket, zum anderen ein Optimalschutz Paket. Die Ausgabe wird um einen zusätzlichen Punkt ergänzt, „keine Versicherung“/. Die Auswahl der Versicherungsart lässt sich über die in HTML definierten Radiobuttons treffen, es ist immer nur eine Auswahl möglich. Um höheren Komfort zu bieten, bleibt die Auswahl bei einem erneuten Aufruf der Seite bestehen. Folgt eine Bestätigung und der Versuch auf die nächste Seite zu gelangen werden nämlich alle Eingaben, nach einer Prüfung, in die Session geschrieben. Die Ausgabe der Punkte ist an Bedingungen geknüpft, welche auf die Einträge in der Session eingehen. Dort sind Angaben über die zuletzt gewählte Versicherung hinterlegt. Lässt sich der Wert aus der Session einer bestimmten Auswahl zuordnen wird diese vorausgewählt. Sollte kein passender Wert existieren liegt die Vorauswahl auf „Keine Versicherung“. Eine Ausgabe der Zusatzleistungen geschieht in der Reihenfolge nach inbegriffenen Leistungen, Leistungen die bezahlt werden müssen und freiwilligen Leistungen. Pflicht-Leistungen werden während des Abrufs in der Session gespeichert, da sie ohnehin zu bezahlen sind. Eine Ausgabe von zusätzlich buchbaren Leistungen ist schwieriger. Sie sind mit Checkboxes ergänzt über die sich eine Auswahl treffen lässt. Rücksicht muss auch auf eine mögliche Vorauswahl genommen werden, wenn die Seite neu geladen wird. Dazu importiert das Script alle Einträge aus dem Bereich **zusatz** der Session und initialisiert einen Schalter, der Auskunft gibt, ob eine Leistung schon einmal gewählt wurde. Sollte eine Leistung aus der Datenbank mit einer Leistung der Session übereinstimmen liegt der Schalter auf „1“, sonst „0“. Er bestimmt ob der Punkt bereits angewählt sein soll oder ihn der User standardmäßig ohne Vorauswahl zu Gesicht bekommt.

Zusatzleistungen sind ein Teil der Übersicht. Viel wichtiger sind Preise und Zahlungsdetails. Die Ausgabe ist in zweierlei Form möglich, mit oder ohne Angabe einer Anzahlung. Die Anzahlung ist nur dann präsent, wenn der Zeitraum zwischen Buchung und Anreise nicht zu gering ausfällt. Sonst setzt Interhome auch den Wert für Anzahlung auf den Vollpreis und gibt als Datum „0001-01-01“ an.

Im letzten Teil muss noch eine Zahlungsart ausgewählt werden. Eine Zahlung auf Rechnung ist an die Bedingung geknüpft, dass der Zeitraum zwischen Buchung und Anreise größer als 29 Tage ist.¹²² Fällt der Abstand kleiner aus wird diese Option angezeigt, aber eine Auswahl ist nicht möglich. Der Abstand lässt sich wieder aus der Differenz zwischen Anreisedatum und Buchungsdatum berechnen und steuert die Belegung einer Variable. Bei weniger als 29 Tagen enthält sie die Anweisung die Auswahl zu deaktivieren. Gleichfalls kommt es wie schon bei der Versicherung zu einer Vorauswahl, wenn die entsprechende Zahlungsart schon in der Session hinterlegt ist. Dabei darf die Auswahl „Auf Rechnung“ nicht getroffen werden, wenn sich das Datum ändert und der Zeitraum zwischen Buchung und Anreise unter 29 Tage gefallen ist.

Einige Anbieter geben danach noch Informationen aus. Meist handelt es sich um Rückerstattungskosten bei Nichtantritt der Reise oder die AGB's. Nur wenn sie bestätigt werden ist ein Weiterkommen möglich.

Eingabefelder, Auswahlmenüs und Boxen werden mit Hilfe einiger Javascript Funk-

¹²² Interhome AG 2008b, 18ff und Lock 2008a, 2

tionen auf Korrektheit überprüft. Treten Fehler auf, folgen Fehlermeldungen an den entsprechenden Stellen im Formular und der Cursor springt in das fehlerhafte Feld. Die wichtigere Prüfung folgt jedoch erst Server-seitig beim Absenden des Formulars und bringt den User bei Falscheingaben wieder auf dieses zurück.

9.7.2. Funktion „Zusatzleistungen Wolters“

Wie der Name schon vermuten lässt, ist diese Funktion für die Ausgabe von Preisen und Leistungen der Wolters Objekte zuständig. Wie von Funktionen in einer Klasse bekannt müssen auch hier einige Werte an die Funktion übergeben werden. Dazu zählen u.a. Anbieterkennung und Referenzcode von Objekt und Kategorie sowie Anreisedatum und Aufenthaltsdauer. Einige Prozeduren sind mit denen aus „Zusatzleistungen Interhome“ identisch. Es lassen sich hier z.B. die Ermittlung einer maximalen Belegung aufzählen. Unterschiede bringt schon die Erfassung von Haustieren in der Buchung mit sich. Wolters bietet 2 Möglichkeiten an. Zum einen lässt sich ein Haustier über die Zusatzleistungen wählen, zum anderen kann es schon als Merkmal für das Objekt angegeben sein¹²³. Über die Zusatzleistungen lässt sich nur 1 Haustier wählen, als Merkmal sind bis zu 3 erlaubt. Die Buchungsschnittstelle muss darüber informiert sein, welche Variante vorliegt. Deshalb gibt es für Haustiere 2 Felder im Array und eine Steuerung, die die jeweils gültige Variante wählt. Dies fängt bereits mit der Abfrage von Merkmalen aus der Datenbank an. In einer Variable wird die Anzahl an Haustieren erfasst. Es sind laut Wolters Dokumentation 1 bis 3 Haustiere möglich¹²⁴. Für den Fall das keine Anzahl angegeben ist, liegt der Wert auf „0“. Buchungshinweise spielen bei diesem Anbieter eine wichtige Rolle. Jeglicher Hinweis muss akzeptiert werden, bevor es mit der Buchung weiter geht¹²⁵. Hierfür folgt auf den Hinweis eine Checkbox. Liegen keine Hinweise vor erhält sie automatisch ein Häkchen. Sonst muss immer der User bestätigen. Aus Sicherheitsgründen merkt sich das Script diese Einstellung beim erneuten Aufruf der Seite nicht. Der User muss deshalb jedes mal erneut bestätigen. Dies gilt im übrigen auch für die AGBs. Hinweise können mit oder ohne Datumsangabe auftreten. Ist ein Datum vorhanden, ist es auch in der Ausgabe zu sehen. Meist handelt es sich um Zeiträume für die der Hinweis gilt. Kritisch sind Hinweise auf Kautio. Da dem User im späteren Verlauf ein Endpreis angezeigt werden soll, der nahezu dem entspricht was ihn an Kosten durch Zusatzleistungen, Kurtaxen und Kautionen entsteht, muss der Kautionspreis extrahiert werden. Wolters bietet ihn nicht extra an. Suchkriterium sind **Kautio**, **EUR** und **Euro**. Die Wahrscheinlichkeit ist groß, dass Angaben wie Euro oder EUR ein Preis voran steht. PHP stellt Stringfunktionen zur Bearbeitung von Zeichenketten bereit. Durch Funktionen wie *strstr*, *trim* und *substr* kann der Preis aus dem Buchungshinweis extrahiert und in die Session übernommen werden.

Die Abfrage und Verarbeitung von Zusatzleistungen aus der Datenbank greift auf eine Funktion zu, die auch für Interhome Objekte anwendbar ist. Sie erfasst jedoch alle Zusatzleistungen und unterscheidet nicht nach Kategorien. Dadurch entsteht kein

¹²³ Lock 2008a, 2

¹²⁴ Lock 2008c, 1

¹²⁵ Lock 2008c, 4

Nachteil, denn Wolters unterteilt die Leistungen nicht in mehrere Varianten. Jegliche Leistungen können freiwillig zugebucht werden. Dadurch vereinfachen sich Verarbeitung und Ausgabe erheblich. Die Ausgabe erfolgt über eine *Foreach* Schleife. Vor dem Aufruf kommt es zur Initialisierung einer Zählvariable, eines Schalters für Haustiere als Leistung und einem Schalter der die Vorauswahl steuert. Sollte „Haustier“ als Leistung auftreten wird der Schalter auf 1 und damit aktiv gesetzt. Als Ausgabe dienen Pull-Down Menüs, in der HTML Sprache *select*¹²⁶ genannt. Eine Zählvariable *k* stellt den Namen des select Bereiches dar. Für jedes Feld innerhalb des Bereiches ist der Zähler *i* zuständig. Jedes Feld erhält als Wert Informationen über die Zusatzleistung und ihre Anzahl die gebucht werden soll. Felder mit der Anzahl 0 benötigen keine Angaben zur Leistung, weshalb sie weggelassen werden. Pro Leistung ist eine Angabe möglich, die der max. Belegung des Hauses entspricht¹²⁷. Auf ein Haus mit max. 6 Personen lassen sich z.B. 6 Handtücher wählen. Ausnahmen bilden sowohl Haustier und Endreinigung. Bei beiden ist nur max. 1 Posten zubuchbar. Dies trifft bei Haustieren jedoch nur dann ein, wenn es explizit als Zusatzleistung vorliegt. Taucht es als Merkmal auf sind max. 3 Haustiere möglich. Wenn ein User auf die Seite zurückkommt, z.B. wegen fehlerhafter Eingaben, darf keine Eingabe zurückgesetzt sein. An dieser Stelle kommt der 2. Schalter zum Zug. Vor dem Aufbau des Auswahlmenüs wird die Session nach Zusatzleistungen durchsucht. Sollte die aktuelle Leistung mit einer aus der Session übereinstimmen, bekommt der Schalter den Wert 1 und ist damit auf aktiv gesetzt. In welchem Umfang der User die Leistung einmal ausgewählt hat lässt sich ebenfalls anhand der Session ermitteln. Er dient zur Rekonstruktion der Auswahl. Da der Schalter aktiv ist liegt als Standardauswahl der zuletzt eingestellte Wert vor.

Unter den Zahlungsarten ist die Zahlung auf Rechnung ebenfalls möglich. Hierbei gilt ähnliches wie für Interhome. Die Zahlungsart kann nur gewählt werden, wenn Anreise und Buchung mindestens 6 Wochen auseinander liegen¹²⁸. Es sollte klar sein, dass hier eine ähnliche Prozedur angewendet wird, wie sie für Interhome existiert. Lediglich die Zeitspanne musste angepasst werden. Gleichfalls ist hier auch die Zahlungsart vorausgewählt, welche beim letzten Aufruf des Formulars eingestellt wurde. Im Standardfall ist *Bankeinzug* gewählt falls es noch keine Auswahl gab.

9.8. Zahlungsangaben

Am Beginn steht die schon bekannte Kontrolle der Eingaben an. Dafür ist die *buchen_varcheck.php* zuständig. Sie ist in den 2. Teil der Buchung inkludiert und bearbeitet alle Eingaben die im 1. Buchungsformular getätigt wurden. Nur wenn alle Eingaben korrekt sind läuft das Hauptscript weiter. Bei Fehlern kommt es zur Weiterleitung auf den 1. Buchungsschritt. Durch Auswertung des Parameters für Zahlungsart ist das Script in der Lage die richtigen Formularfelder für weitere Eingaben darzustellen. Zahlungstypen sind mit 1 bis 3 Codiert. Die 1 steht für Bankeinzug, 2 die Kreditkartenzahlung und 3

¹²⁶ Skop 2007, HTML - Formulare - Auswahllisten, Radio-Buttons und Checkboxes

¹²⁷ Lock 2008c, 5

¹²⁸ Lock 2008a, 2

für Zahlung auf Rechnung. Alle Eingaben durchlaufen zu Überprüfungszwecken einige Javascript Funktionen. Die spätere Kontrolle auf dem Server durch PHP Funktionen ist obligatorisch. Eine Abfrage der Kreditkarteninformationen erfordert ein zusätzliches Query auf die Datenbank. Da jeder Anbieter die Namen der Kreditfirmen anders abkürzt, (vgl. dazu ¹²⁹ und ¹³⁰) existiert eine Tabelle, wo alle Abkürzungen mit dem zugehörigen Namen des Anbieters aufgeführt sind. Außerdem bietet kaum ein Anbieter die selben Kreditkartentypen an wie ein anderer. Das Auswahlmenü für den Kreditkartentyp wird immer passend zum Anbieter erzeugt. Hat der User schon in einem früheren Aufruf eine Auswahl getroffen, wird diese bei einem erneuten Aufruf selbstverständlich wiederhergestellt. Dies gilt ebenso für weitere Daten. Um den Komfort zu erhöhen ist die Angabe des Verfallsdatums der Karte über 2 Auswahlmenüs geregelt. Im 1. Menü folgt die Jahresangabe. Sie wird dynamisch generiert und umfasst, beginnend mit dem aktuellen Jahr, einen Zeitraum von 15 Jahren. Da sich Monate nicht ändern, liegen sie als statische Angaben vor. Vorausgewählt ist der aktuelle Monat des aktuellen Jahres. Wählt der User die Bezahlung auf Rechnung, muss er im Zahlungsformular nur mit einem Häkchen akzeptieren.

The image shows a web form titled "Zahlungsangaben" (Payment Information). Inside, there is a section titled "Kreditkartenzahlung" (Credit Card Payment) enclosed in a dashed box. The form contains the following fields:

- Karte:** A dropdown menu with "Mastercard" selected.
- Kartennummer:** A text input field.
- Verifizierungsnummer:** A text input field.
- Verfallsdatum:** Two dropdown menus showing "2009" and "2".
- Kreditkarteninhaber:** A text input field.
- Name:** A text input field.
- Vorname:** A text input field.

At the bottom left of the dashed box, there is a button labeled "weiter".

Abbildung 9.12.: Eingabe von Zahlungsinformationen, hier Kreditkartenzahlung

Die Überprüfung der Zahlungsangaben findet in 1. Instanz durch Javascript statt. Hier besteht der Vorteil, dass die Seite nicht neu geladen werden muss und der Cursor in die Felder springt, welche fehlerhafte Eingaben aufweisen. Natürlich folgt auch hier eine Kontrolle auf dem Server durch PHP um ungültige Angaben zu verhindern.

¹²⁹ Interhome AG 2008a, 12f

¹³⁰ Lock 2008a, 2

9.9. Eingabeüberprüfung Buchungsmaske

Wie bereits bei der Suchmaske oder der Objektübersicht gibt es hier ebenfalls für jeden Buchungsschritt ein vorgeschaltetes Kontrollscript. Das Prinzip entspricht eben diesem. Jede User-Eingabe wird auf Validität geprüft und in der Session gespeichert. Fehler führen zu einem Rücksprung auf die Seite, von welcher die Informationen abgeschickt wurden. Gleichzeitig sind dort alle Felder mit einem Hinweis markiert die Fehler enthalten.

9.10. Darstellung der Angaben

Wichtig ist die Anzeige aller getroffenen Eingaben bevor die Buchung abgeschickt wird. Für den User besteht die Möglichkeit seine Angaben zu überprüfen und g.g.f. Korrekturen vorzunehmen.

Wie bereits aus dem Formular für die Erfassung personenbezogener Daten bekannt, kommt es auch in diesem Fall zur Ausgabe anbieterspezifischer Daten. Ihre Verarbeitung geht ähnlich von statten, d.h. jeder Anbieter bekommt seine eigene Funktion. Betroffen sind davon allerdings nur die Zusatzleistungen. Sie sollen nicht nur aufgelistet und der Preis pro Leistung und deren Berechnung genannt werden. Für den User ist es wichtig, eine Preisvorstellung von dem zu bekommen, was ihn erwartet. Dazu zählen eben auch alle zusätzlichen Leistungen. Die Berechnung ähnelt sich für die beiden exemplarisch genannten Anbieter. Anhand von Personenanzahl, Aufenthaltsdauer, Preis und Berechnungsart werden die Gesamtkosten pro Leistung ermittelt. Es folgt sogar eine Unterscheidung zwischen der Anzahl an Erwachsenen Reiseteilnehmern und Kindern, denn einige Leistungen wie Hochstuhl und Kinderbett beziehen sich auf mitreisende Kinder. Beide Anbieter stellen die Berechnungsarten als beschreibenden Text dar. Für die Auswertung wurde deshalb in der entsprechenden Funktion ein Array angelegt was alle Berechnungsarten enthält. Der Abruf der Zusatzleistungen führt dazu, dass ein Vergleich des Terms mit dem Array stattfindet. Es kommt zu einer Fallunterscheidung. Nun wäre es auch möglich alle Berechnungsarten direkt in jedem Fall anzugeben. Um eine gewisse Flexibilität zu gewähren, gibt es den Umweg über das Array. Der Vergleich findet mit Hilfe von *preg_match* statt. Da es z.B. bei Interhome 5 Berechnungsarten gibt, müssen 5 Fälle und ein Standardfall berücksichtigt werden¹³¹. Innerhalb der Fälle kommt es zur entsprechenden Berechnung des Preises. Sollte keine Berechnungsart genannt sein, gilt der Standardfall - die Übernahme des Ausgangspreises in die Angaben. Sind alle Leistungen bearbeitet, folgt außerdem die Angabe eines Gesamtpreises. Nebenbei durchlaufen alle Preise eine Prüfung auf Währungsart. Nicht jeder Inserent bietet sein Objekt in der europäischen Einheitswährung an. Britische Vermieter legen beispielsweise Wert auf Preisangaben in britischem Pfund. Da jeder Preis mit einem Währungscode ausgestattet ist, lässt sich über diesen die richtige Währung und damit ein Wechselkurs zuordnen. Wechselkurse liegen in einer Tabelle der Datenbank vor. Über sie findet eine zusätzliche Umrechnung in Euro statt. Da solch ein Schritt immer in der selben Art und Weise

¹³¹ Interhome AG 2008a, 17f

Adressdaten Zahlungsangaben Angaben prüfen Buchung bestätigen	
Ferienhaus in Langfurth Bayerischer Wald, Deutschland Objekt: DE8351.100.2	
	
Aufenthalt vom 07.03.2009 für 14 Tage Objektpreis: 518 EUR Reise Kostenschutz: 0 EUR Zusatzleistungen: 3 EUR	
Ihre Daten	
Anrede: Herr Name: Max Vorname: Mustermann Strasse: Musterstr 123 Postleitzahl: 01234 Ort: Musterhausen Herkunftsland: Deutschland Telefon: 0123456	Fax, Tel2: 0123457 Handy: E-Mail: max@muster.de Anz. Reisender: 3 Davon Kinder: 1 AGBs: Bestätigt
Zusatzleistungen und Zahlungsdetails	
Zahlung mit Kreditkarte Karte: VisaCard Kartennummer: 3452345629834521 Verifizierung: 123 Gültigkeit: 10/2009 Karteninhaber: Mustermann Max	Preis des Objektes: 518 EUR Versicherung: 0 EUR Objekt Gesamt: 518 EUR Verpflichtende Zusatzleistungen: Freiwillige Zusatzleistungen: Toilettenwische, 3.00 EUR 1x, pro-Person Gesamt: 3.00 EUR Buchungsgebühr: 50 EUR Gesamtkosten Zusatzleistungen: 3 EUR Buchungskosten: 568 EUR
Die Buchungskosten beziehen sich auf den Objektpreis. Verpflichtende Zusatzleistungen wie z.B. Kautionen und Zusätzlich gebuchte Leistungen sind vor Ort zu begleichen.	
Objekt- und Reisedaten	
Gebuchtes Objekt: DE8351.100.2 Anreisedatum: 07.03.2009 Aufenthaltsdauer: 14 Tage Angaben korrigieren	
Buchung	

Abbildung 9.13.: Ausgabe der erfassten Daten

abläuft, wurde er als Funktion ausgelagert. Jede Leistung ist dann in der Ausgabe sowohl mit der Landeswährung als auch einem Preis in Euro aufgeführt. Selbiges gilt für den Gesamtpreis.

Der Ablauf für Wolters Objekte ist dem gerade genannten sehr ähnlich. Unterschiede gibt es in den Berechnungsarten und deren Beschreibung. Auch entfällt die Währungsumrechnung, da Tui-Wolters alle Objekte standardmäßig mit der europäischen Währung listet.¹³²

Die Angabe personenbezogener Daten ist in allen Fällen gleich und bedarf keiner Unterscheidung. Es folgt lediglich die Ausgabe der jeweiligen Werte aus der Session. In einige Angaben wie beispielsweise Bestätigung von AGBs oder Buchungshinweisen ist ein Schalter eingebaut. Sollte davon etwas nicht akzeptiert worden sein, sorgt er

¹³² Lock 2007b, 4ff



Abbildung 9.14.: Ablauf Datenausgabe

dafür, dass ein Absenden und Bestätigen der Buchung unmöglich ist. Ebenso simpel ist die Darstellung der Zahlungsmodalitäten. Anhand der Codierung folgt ein Aufruf der entsprechenden Daten.

Im folgenden ist es dem User möglich, seine Daten entweder zu korrigieren oder die Buchungsangaben abzusenden. Anhand der Anbieterkennung ist definiert welches Script für die Aufbereitung und den Versand der Daten angesprochen wird.

9.11. Übermittlung der Buchungsdaten

Jeder Anbieter hat andere Schnittstellen und Definitionen zur Übertragung der Daten¹³³. Die Aufbereitung läuft jedoch immer nach dem gleichen Prinzip. Alle übertragungsrelevanten Informationen werden aus der Session in aussagekräftige Variablen übernommen.

¹³³ Interhome 2008b, 8ff und Lock 2008a, 1ff

Dabei dienen die Bezeichnungen der Anbieter als Ausgangspunkt. Bezeichnet der Anbieter z.B. das Anreisedatum als Checkin heißt auch die Variable in der der Wert erfasst wird „checkin“/. Für einige Angaben ist die einfache Übernahme nicht möglich. Im Vorfeld müssen dann u.U. mehrere Eintragungen miteinander verglichen und daraus eine Aussage getroffen werden. Als Beispiel dient hier die Buchung zusätzlicher Betten für ein Haus. Interhome bietet Zusatzbetten, aber auch Kinderbetten an. Daraus ergeben sich 3 Möglichkeiten. Es kann entweder eines von beiden gewählt sein oder beide werden gebucht. Für jede dieser Möglichkeiten hält Interhome einen Code vor. Hier ist ein Vergleich beider Werte nötig, um die richtige Aussage und damit den richtigen Code anzugeben.

Listing 9.1: Zusammenfassung von Werten

```

1 {
2   ...
3   if ($SESSION['zusatz'][$i][0] == 'Zusatzbett') {
4     $extrabed1 = 'A';
5     $extrabed = 'A';
6   }
7
8   if ($SESSION['zusatz'][$i][0] == 'Kinderbett') {
9     $extrabed2 = 'C';
10    $extrabed = 'C';
11  }
12
13  if ($extrabed1 == 'A' AND $extrabed2 == 'C') {
14    $extrabed = 'B';
15  }
16  ...
17 }
```

Manche Angaben sind von einem konkreten Wert oder einer Beschreibung in einen boolschen Ausdruck zu konvertieren. Als Beispiel kann die Mitführung eines Haustieres genannt werden. Der Anbieter möchte nur wissen, ob ein Haustier mitreist. Also wird in der Session nach einem Eintrag *Haustier* gesucht. Dabei spielen alle weiteren Angaben zum Eintrag keine Rolle. Hauptsache er ist vorhanden. Dann kann die Frage mit einem Y beantwortet werden. Liegt kein Eintrag vor, ist die Variable mit N besetzt.

Hinzu kommen Standardangaben wie beispielsweise ID des Partners, des Verkaufsbüros und weitere. Die vorliegende Zahlungsart entscheidet darüber welche der Variablen für Zahlungsangaben belegt werden und welche keinen Wert zugewiesen bekommen. Eine Zahlung per Kreditkarte bedingt, dass Angaben für Rechnung oder Bankeinzug nicht gegeben sind.

Sind alle Werte übernommen wird bei Interhome Objekten eine Verbindung zur SOAP Schnittstelle aufgebaut. Abhängig von der Zahlungsart werden unterschiedliche Informationen, die Zahlungsart betreffend, versandt. Als Rückgabe liefert Interhome die Währung, Objektprice, Buchungsgebühr, ein Datum bis wann die Anzahlung erfolgen muss inkl. Anzahlungspreis und den Gesamtbetrag.¹³⁴

In letzter Instanz fügt das Script einen Eintrag in die Datenbank ein. Hierbei werden Angaben zur Person, Buchungszeit und Buchungsstatus erfasst. Dieser Schritt soll dafür sorgen, dass es bei eventuellen Rückfragen nicht zu Unklarheiten kommt und ein Beleg

¹³⁴ Interhome AG 2008b, 18ff

über die Buchung vorhanden ist. Der User bekommt nach dem Abschluss der Buchung die Rückgabewerte vom Anbieter zu Gesicht und erfährt so, ob eine Buchung erfolgreich war.

Tui-Wolters benutzt eine andere Schnittstelle für die Datenübermittlung. Sie basiert auf der HTTP-Methode GET. Im Vorfeld findet nicht nur eine Übertragung von Werten aus der Session in Variablen statt, sondern auch eine URL kompatible Zeichenkodierung. Hierfür stellt PHP die Funktion *urlencode*¹³⁵ bereit. Wird dieser Umstand nicht beachtet, kommt es zu einer fehlerhaften Übertragung und die Buchung kann nicht bearbeitet werden. Die Abkürzungen der Länder müssen rekodiert werden, da Wolters andere als die üblichen benutzt. Hierfür sind im Script zwei Arrays angelegt, eines mit den standardmäßigen, das andere mit Wolters Codierung. Die Positionen einzelner Länder mit den jeweiligen Abkürzungen ist bewusst identisch. Innerhalb einer Schleife sorgt ein Vergleich dafür, die Position einer Abkürzung im Array zu ermitteln. Da diese nun mit den Positionen für Abkürzungen von Wolters Objekten gleich ist, muss nur noch der entsprechende Eintrag aus dem zweiten Array gelesen werden. Damit ist die Konvertierung abgeschlossen.

Listing 9.2: Konvertierung Wolters Landescodes

```

1 {
2   ...
3   $land_array = array('AT', 'AU', 'BE', 'BG', 'CA', 'CH', 'CS',
4     'CY', 'CZ', 'DE', 'DK', 'ES', 'FIN', 'FR', 'GB', 'GR', 'HR',
5     'HU', 'IE', 'IT', 'LT', 'MT', 'NL', 'NOR', 'PL', 'PT', 'RU',
6     'SE', 'SI', 'SK', 'TR', 'US');
7   $landwo_array = array('A', 'AU', 'B', 'BG', 'CA', 'CH', 'CS',
8     'CY', 'CZ', 'D', 'DK', 'E', 'FIN', 'F', 'GB', 'GR', 'HR', 'H',
9     'IRL', 'I', 'LT', 'MT', 'NL', 'N', 'PL', 'P', 'RU', 'S', 'SI',
10    'SK', 'TR', 'US');
11   foreach ($land_array AS $land_kz) {
12     if ($SESSION['person'][6] == $land_kz ) {
13       $land = $landwo_array[$i];
14     }
15   }
16   ...
17 }
```

Ein weiterer Sonderfall sind die beiden Möglichkeiten ein Haustier mit zu buchen. Hierfür wurde im Vorfeld definiert, dass ein Haustier welches aus den Merkmalen hervorgeht im Sessionarray für Zusatzleistungen immer auf Feld 4 liegt. Sind auf diesem Feld Werte hinterlegt folgt eine weitere Auswertung der Daten. Dazu gehört u.a. die Anzahl an Vierbeinern welche mitreisen. Ist nur ein Haustier gegeben, bekommt die Wolters Variable **pet** ein **yes**. Für mehr als ein Haustier wird die entsprechende Anzahl an Haustieren angegeben. Liegt das Haustier dagegen als Zusatzleistung vor bekommt **pet** den Wert **no**. Zusatzleistungen sind intern codiert und lassen sich über die Tabelle *ref_zusatzleistungen* rekonstruieren. Wolters sieht ebenfalls eine Codierung vor welche gleichermaßen jeder Leistung in der Tabelle zugeordnet ist. Jede übermittelte Leistung setzt sich aus dem Objektcode, Leistung und Anzahl an gebuchten Einheiten zusammen.¹³⁶ Hinzu kommen persönliche Daten und Zahlungsangaben. Sind alle Werte als URL aufbereitet, können

¹³⁵ Achour 2009, Funktionsreferenz - Sonstige Grunderweiterungen - URLs - urlencode

¹³⁶ Lock 2008a, 2

sie versendet werden. Da Wolters Bestätigungsinformationen als XML an den Absender zurücksendet wurde der Versand an die PHP Funktion *simplexml_load_file* geknüpft. Ob es sich bei der Antwort wirklich um ein XML Objekt handelt, lässt sich über die Funktion *is_object*¹³⁷ aus den PHP Bibliotheken bestimmen. Trifft die Bedingung zu lassen sich die empfangenen Daten auswerten. Sollten Fehler auftreten sieht Wolters 2 Fehlerfälle vor¹³⁸. Kommt es bereits während des Datenversandes zu Problemen, z.B. durch Unterbrechung in der Übertragung oder fehlende bzw. ungültige Nutzereingaben trifft der 1. Fall zu. In der Antwort steht dann „Incomplete Booking Data“. Im 2. Fall liegt der Fehler auf Seiten des Anbieters z.B. weil das Haus in der Zwischenzeit schon von einem anderen User gebucht wurde oder es anderweitig zu Problemen beim Anbieter kommt. Die daraus resultierende XML Datei ist schon anders strukturiert als im 1. Fall und gibt deshalb schon Hinweise auf einen Fehler, der nicht durch den User oder das Ferienhaus.de System verursacht wurde. Auch kommt es zu einer anderen Fehlermeldung als im 1. Fall. In den XML Daten sind zudem weitere Informationen über den Buchungsstatus gespeichert. Es handelt sich um zwei Punkte, **bookingstate** und **buchungsstatus** ist bei beiden der Satus auf OK gesetzt war die Buchung erfolgreich. Ist nur einer von beiden nicht mit OK gekennzeichnet war die Buchung nicht erfolgreich. Für erfolgreiche Buchungen liegen ebenfalls eine Buchungsnummer und Gesamtpreis in der Antwort vor. Im Script, welches die Buchungsanfrage an Wolters sendet, ist ebenfalls ein Schalter definiert. Treten Fehler auf bekommt dieser den Wert 1, sonst 0. Er ist für die spätere Eintragung des Buchungsvorganges in die Datenbank wichtig. Der User wird darüber informiert welcher Status vorliegt. Für eine erfolgreiche Buchung kommt es neben der Statusmeldung zur Ausgabe der Werte aus der Antwort des Anbieters, also u.a. Buchungsnummer und Gesamtpreis. In der Datenbank wird ein Eintrag über den Buchungsvorgang erzeugt. Der User kann nun andere Seiten aufrufen oder auf die Suchmaske zurückkehren.

Damit sind alle Schritte von der Suche über die Auswahl, Detailansicht und Buchung abgeschlossen. Tests mit dem System waren erfolgreich. Allerdings sind noch einige Änderungen und Korrekturen vorzunehmen. Es ist anzuraten für den Datenversand eine verschlüsselte Verbindung, z.B. über SSL zu benutzen.

¹³⁷ Achour 2009, Funktionsreferenz - Variablen- und typbezogene Erweiterungen - Variablenbehandlung - *is_object*

¹³⁸ Lock 2008a, 3ff



Abbildung 9.15.: Ablauf Datenversand

10. Erreichte Ziele

10.1. Datenaggregation

Der Abruf von Anbieterdaten konnte automatisiert werden. Die Verbindung mit den FTP Servern wird selbstständig aufgenommen und die Daten herunter geladen. Liegen die Daten in gepackter Form vor erfolgt zusätzlich das Entpacken.

Die Verarbeitung von Daten läuft ebenfalls voll automatisiert ab. Dies umfasst die Aufsplittung, Referenzierung, Recodierung und Neuzusammensetzung der gelieferten Datensätze. Ebenfalls enthalten ist das Update betreffender Tabellen in der Datenbank.

Die Protokollierung der Vorgänge wurde nachträglich eingefügt. Dabei mussten ein paar Strukturen in der Verarbeitungsreihenfolge geändert werden, um zuverlässige Abbruchsequenzen gewährleisten zu können. Dies bezieht sich auf Vorgänge, die auf eine vorherige Aktion angewiesen sind und bei Fehlern in dieser weitere Fehler produzieren würden.

Fehler wie beispielsweise ein missglückter Zugriff auf Logfiles oder einer gescheiterten Abholung der Daten führen zum Versand einer E-Mail an den technischen Ansprechpartner.

Die Referenzierung der Geografischen Anbietercodierungen auf das interne System läuft nahezu fehlerfrei ab. In Ausnahmefällen treten jedoch Referenzierungsfehler auf, dessen Ursache noch nicht gefunden wurde. Betreffende Daten werden jedoch nicht in die Datenbank übernommen und landen interessanterweise in einem zweiten Versuch mit korrekter Referenzierung in der Datenbank.

Probleme mit der Größe einiger Daten konnten überwunden werden. Es lassen sich dadurch alle gelieferten Daten der Anbieter in die Datenbank übernehmen.

10.2. Datenbank

Der Aufbau nach dem vorliegenden Konzept ist erfolgt. Für eine bessere Unterscheidung sind die Tabellennamen mit einem vorangestellten Typ gekennzeichnet. Die Struktur der einzelnen Tabellen lies sich aus den Strukturen der Anbieterdaten gewinnen. Dabei flossen Angaben aller Anbieter ein, so dass einige Tabellen z.T. mehr Angaben aufnehmen können als einzelne Anbieter liefern. Einige nicht vorher festgelegte Tabellen entstanden in denen u.a. Angaben zu Kreditkartentypen und Wechselkursen zu finden sind. Die Zusatzleistungen wurden durch eine Tabelle ergänzt in der Referenzierungen zur Codierung der Leistungen einzelner Anbieter enthalten sind. Sie stellt damit einen Vorgriff auf das System dar, was später evtl. als Ersatz zur aktuellen Referenzierung geografischer Daten angedacht ist.

10.3. Buchungssystem

Ein Buchungssystem wurde aufgebaut, bestehend aus Suchmaske, Objektübersicht, Detailansicht, Merkliste, Vakanzabfrage und eigentlicher Buchung. Alle getätigten Eingaben werden temporär gespeichert und gehen mit Beenden des Browsers auf Clientseite verloren. Die Zwischenspeicherung ist auch dann möglich, wenn der Client nur mit einem eingeschränkten System arbeitet. Eine Benutzung der Merkfunktion ist dabei auf diesen Systemen u.U. nicht möglich. Dies liegt in der Natur der Sache, da eine Zwischenspeicherung von Werten über eine Session, die Erfassung von Objekten in der Merkliste über Cookies realisiert wird.

Die Suche in der Suchmaske ist dabei über die geforderten Wege möglich. Eine Unterteilung nach einfacher und erweiterter Suche findet statt. Die angeführten Möglichkeiten der Auswahl sind in beiden Sucharten integriert. Eine Auswahl des Reiseziels erfolgt über Prozesse die es ermöglichen relevante Teile der Seite nachzuladen ohne die Seite komplett neu aufzubauen. Auf eingeschränkten Systemen wird automatisch eine Alternative Suchmaske mit Verzicht auf dieses Merkmal geladen um die Funktionalität zu gewährleisten.

Die Objektübersicht listet standardmäßig einen Umfang von 10 Objekten pro Seite auf. Die Möglichkeit deren Anzahl auf einer Seite zu erhöhen ist gegeben. Das seitenweise Blättern mit Angabe von max. 10 Seiten wurde ebenfalls integriert. Sollten mehr Objekte darzustellen sein als auf 10 Seiten passen, läuft die Seitenwahl bei Auswahl einer höheren bzw. niedrigeren Seitenzahl mit. Dadurch sind auch Objekte auf weiter hinten liegenden Seiten erreichbar. Die Ausgabe von Objektinformationen umfasst die Geforderten Punkte. Dies sind im einzelnen Kurzbeschreibung, Merkmale, Preise, Bild und Objektcode. Eine Top Navigation ist vorhanden. Sie verweist auf Suchmaske und Merkliste.

In der Detailansicht werden Lage des Objektes, Bildergalerie und Zugehörige Merkmale, Bewertungen und Beschreibungstexte dargestellt. Verweise auf die Belegungsübersicht mit anschließender Buchung oder Aufnahme des Objektes in die Merkliste sind integriert. Auch hier existiert eine Top Navigation mit Verweis auf die Suchmaske, Übersichtsseite und Merkliste. Sollte ein Aufruf der Detailansicht direkt aus der Suchmaske erfolgen entfällt der Verweis auf die Übersicht. Der Bildergalerie ging eine Variante mit Blätterfunktion voraus. Sie wurde jedoch vom Auftraggeber verworfen. Eine Präsentation mit bis zu 16 Vorschaubildern erschien in Sachen Kundenanreiz sinnvoller als eine Vorschau von 3 Bildern mit durchblättern. Gleichzeitig stellte die neue Variante einen geringeren Entwicklungsaufwand dar.

Eine Merkliste wurde aufgebaut. Sie ist, wie gefordert, in der Lage Objekte aufzunehmen, anzuzeigen oder einzelne bzw. alle Objekte zu löschen. Die Speicherung basiert auf Cookies welche auf dem Rechner des Users hinterlegt werden. Sind Cookies gesperrt ist die Merkliste nicht nutzbar. Die Merkliste enthält eine Begrenzung auf 25 Objekte. Ihr Aufruf bei Eintragungen läuft, wie es gewünscht wurde, im Hintergrund ab.

Die Belegungsübersicht ist ebenfalls entstanden. Sie beherrscht die Unterscheidung der Anbieter und ruft nachfolgend die entsprechenden Funktionen auf. Die Vakanzen lassen sich vom aktuellen Datum an bis zu einem Jahr im Voraus abrufen. Wie in den

Vorgaben dokumentiert, besteht die Ansicht aus aktuellem und Folgemonat. Anreisetage sind mit einer Preisabfrage verbunden. Sie kommt zum Vorschein, wenn ein Anreisetag angeklickt wurde. Die Auswahl eines daraufhin erscheinenden Preises und Aufenthaltszeitraumes führt zur geforderten Aktualisierung von Anreisetag und Aufenthaltsdauer in der aktuellen Session. Eine gleichzeitige Kontrolle der Datumsangaben sorgt für den zu berücksichtigenden Aspekt, dass der Verweis auf den weiteren Buchungsverlauf nur dann möglich ist, wenn Buchungs- und Anreisetag weit genug auseinander liegen. Auch die Kennzeichnung der Belegung erfolgt wie gefordert farblich. Eine Legende ist aufgeführt.

Die nachfolgende Datenerfassung entspricht ebenfalls den Anforderungen. Personenbezogene Daten werden erfasst. Jeweils zum Anbieter gehörige Zusatzleistungen, AGBs, Hinweise und Zahlungsart tauchen am Ende der Seite auf. Eine Fortführung der Buchung ist nur nach vorheriger Bestätigung von AGBs und Buchungshinweisen möglich. Buchungshinweise sind deshalb zu bestätigen, da sie u.a. für Wolters Objekte Kautionspreise enthalten¹³⁹. Die Anzahl möglicher Personen pro Objekt ist an die max. mögliche Belegung gekoppelt. Die Zahlungsarten richten sich nach den Vorgaben der Anbieter. So fällt eine Zahlung auf Rechnung weg, wenn der zeitliche Abstand zwischen Buchung und Anreise zu gering ist¹⁴⁰. Eine Überprüfung auf JavaScript Basis kontrolliert alle getroffenen Angaben, wenn der User die nächste Seite aufrufen will und markiert alle fehlerhaften Einträge mit einem Hinweis. Auch die Kontrolle auf Server Seite wurde integriert. Sie umgeht das Problem, dass JavaScript eine Client-seitige Sprache ist und vom User im Browser ausgeschaltet werden kann und kontrolliert die Eingaben in jedem Fall.

Eine Erfassung der Zahlungsarten ist ebenfalls Bestandteil des Systems geworden. Eine Kopplung der Ausgabemaske an die Auswahl der Zahlungsart auf der vorherigen Seite ist erfolgt. Zur Anzeige kommen dadurch nur diejenigen Formularfelder welche für die gewählte Zahlungsart nötig sind. Zugleich werden, wie gefordert, pro Anbieter nur die Kreditkartentypen aufgezählt welche der Anbieter unterstützt. Die Auswahl des Verfallsdatum geschieht über Menüs in denen Monat und Jahr des Verfalls wählbar sind. Die angezeigten Jahre erzeugt das Script dynamisch und ist damit immer aktuell.

Die Übersicht mit Darstellung der getätigten Angaben ist ebenfalls entstanden. Der User bekommt hier, wie im Konzept erwähnt, seine Angaben zu sehen und kann ggf. Änderungen vornehmen. Währungsumrechnung beherrscht das Script ebenfalls. Die Darstellung von Währungen abweichend zum Euro findet statt. Angaben über Zusatzleistungen und deren Berechnung mit Hilfe der Berechnungsart gehören zum Repertoire. Zusätzlich sind Angaben zum Objektprice, Kautions, Objektkennung, Anreisedatum und Aufenthaltsdauer enthalten. Dem User ist die Möglichkeit gegeben, auf die erste Seite des Vorgangs zurückzukehren und Änderungen an personenbezogenen Daten, Leistungen und Zahlungsart vorzunehmen. Er kann aber auch seine Daten absenden.

Im Datenversand erfolgt die vorgeschriebene Aufbereitung der Daten nach den Spezifikationen der Anbieter. Interhome erwartet alle Angaben z.B. über die SOAP Schnittstelle während Wolters die Daten über die Http Methode GET entgegen-nimmt. Die Aufbe-

¹³⁹ Lock 2008c, 4

¹⁴⁰ Interhome AG 2008b, 18ff und Lock 2008a, 2

reitung ist gefolgt vom Versand der Daten und einer Auswertung der Rückgabewerte des Anbieters. Der Forderung nach Speicherung des Buchungsvorganges wird Rechnung getragen. Dem User wird zugleich der Erfolg bzw. Misserfolg der Buchung gemeldet.

10.4. Schnittstellen

Eine Kommunikation mit den Schnittstellen der Anbieter wurde in den entsprechenden Stellen des Systems eingebaut. Betroffen davon sind u.a. die Kontrolle des Belegungsstatus eines Objektes, Auskünfte über aktuelle Preise für das Objekt und die Reiseversicherungen.

10.5. Export

Ein Exportscript für die Konvertierung der Daten vom alten System wurde aufgebaut. Es beherrscht die Erzeugung von Stammdaten, bereitet die Merkmale auf und wandelt die Struktur der Vakanzen um. Außerdem exportiert es die Beschreibungstexte und erzeugt eine Datei in der alle Links zu den Bildern enthalten sind. Für jeden Bereich wird eine eigenen Datei angelegt. So ergibt sich eine Struktur die denen der anderen Anbieter entspricht. Das Script passt die Struktur der Daten soweit an, dass die erzeugten Einträge 1:1 in die neue Datenbank übernommen werden können.

10.6. Zusätzliche Scripte

Einige der Scripte übernehmen Aufgaben die mehrfach im System auftauchen und keinem bestimmten Umfeld zuzuordnen sind. Dazu gehören Klassen mit Funktionen für die Kommunikation mit der Datenbank und einer Datenausgabe, Datumsberechnungen und Bildbearbeitung.

Die Bildbearbeitung sorgt dafür, dass Bilder unterschiedlicher Formate auf die vorgegebene Größe reduziert werden. Ihr Format bleibt dabei erhalten. Sie ließ sich in alle Bereiche wo Bildausgaben stattfinden integrieren und leistet dort die geforderte Arbeit. Zu einem späteren Zeitpunkt wurden Input Variablen für die anvisierte Bildgröße integriert, so dass ein flexibler Einsatz für alle Bildgrößen möglich ist.

Eine Reihe von Funktionen sorgt für die Kommunikation mit der internen Datenbank und Ausgabe diverser Daten. Hier erfolgte eine Aufteilung nach den einzelnen Daten. So gibt es Funktionen die alle Zusatzleistungen zu einem bestimmten Objekt ausgeben, die Beschreibungstexte anfordern oder Merkmale abfragen.

Berechnungen von Zeitangaben finden sich ebenfalls als eigene Klasse wieder. Sie enthält Funktionen für die Ermittlung der Anzahl Tage eines Monats, Berechnung von Zeitspannen zwischen zwei Daten und Konvertierung von Datumsformaten.

11. Zukunftsansichten, Verbesserungen

11.1. Backup Strategien

Der Aufbau von Datenbank und Aggregationsroutinen hat viel an Zeit in Anspruch genommen. Ein Crash der Datenbank hätte verheerende Auswirkungen und würde die Arbeit vieler Monate zunichte machen. Deshalb ist nun der nächste Schritt eine Backupstrategie für das System zu entwickeln. Jeden Tag läuft ein Update über die Datenbank und erneuert z.B. die Vakanzdaten der Anbieter. Im Vorfeld sind alle momentan gespeicherten Daten des Anbieters zu sichern. Dabei sollen bis zu 7 Backups vorgehalten sein, für jeden der letzten 7 Tage eines. Zusätzlich ist eine Konsistenzprüfung angedacht. Es besteht die Möglichkeit, dass während eines Downloads die Verbindung abbricht oder der Server nicht mehr reagiert. Dadurch fehlt u.U. ein Teil der Daten. Im Endeffekt fehlen dann in der Datenbank Datensätze. Da allerdings alle Daten in gepackter Form vorliegen, dürfte der Fehler schon beim Entpacken auffallen, da unvollständige Archive nicht komplett entpackt werden können oder der Vorgang sofort abgebrochen wird. Zusätzlich zur Datenbank soll desweiteren ein Backup der Anbieterdaten vorgehalten sein um im Bedarfsfall die Datenbank mit den Backupdaten zu füllen. Gemeint sind hierbei die Archivdateien welche jeder Anbieter auf seinem FTP Server zur Verfügung stellt.

Eine weiterer Vorschlag ist das Anlegen einer zweiten, identischen Datenbank. Sie steht unabhängig von der ersten bereit und ist mit keinem weiteren, öffentlich zugänglichen, System verbunden. In ihr werden die neuen Daten eingelesen und Änderungen an den Daten vorgenommen. Sind alle Schritte erfolgreich, erfolgt ein Backup der ersten Datenbank. Ist dieses beendet, wird sie außer Kraft gesetzt. An ihre Stelle tritt die zweite Datenbank. Sie wird kopiert, so dass beide Datenbanken zu diesem Zeitpunkt identisch sind. Die Kopie kann nun wieder zu Aktualisierungszwecken genutzt werden.

11.2. Stammdatenroutine

Das momentan eingesetzte System erzeugt eine nicht unerhebliche Menge an Redundanzen. Dies liegt darin begründet, dass sich jeder Datensatz aus Land, Region und Ort zusammensetzt. So gibt es z.B. mehrere unterschiedliche Orte, die aber in der selben Region und im selben Land liegen. Die Angaben von Ländern und Regionen wiederholen sich also ständig.

Sinnvoller erscheint es, die Tabellen der Länder, Regionen und Orte mit den internen Codierungen um zusätzliche Spalten zu ergänzen. Jede Spalte ist dabei einem Anbieter zugeordnet und enthält dessen Codierung. Der Datensatz besteht dann aus interner Codierung, Name von Land, Region oder Ort und den Codierungen der Anbieter. Über die relationale Algebra lassen sich alle drei Tabellen verknüpfen. Die Wiederholte Speicherung vom selben Land und Region zu unterschiedlichen Orten entfällt. Gleichzeitig lassen sich Tabellen einsparen. Außerdem ist es leichter zu erkennen, welche Anbieter in den bestimmten geografischen Bereichen tätig sind. Zusätzlich könnte sich die Erkennung von mehrdeutigen Namen mit Bezug auf dasselbe Gebiet vereinfachen, da alle Daten in einer Tabelle liegen.

11.3. Objektcodierung

Es ist eine eigene Codierung aller in der Datenbank vorgehaltener Objekte gewünscht. Dies soll u.a. die Eindeutigkeit in Suchmaschinen steigern. Durch die stetige Aktualisierung der Daten erscheinen immer neue Objekte während andere verloren gehen. Die ursprünglichen Codierungen müssen erhalten bleiben, da ein Anbieter sonst das Objekt nicht wieder erkennt und deshalb eine Buchung nicht bearbeiten kann. Es führt kein Weg an einer Referenztabelle zwischen interner und externer Codierung vorbei. Die Vorgehensweise für Eintragungen dürfte dadurch ähnlich von staten gehen wie sie bereits von den Lagedaten bekannt ist. Allerdings ist im Gegensatz dazu eine eindeutige Referenzierung möglich, da als Referenz der Objektcode dient und dieser eindeutig ist. Leider ist eine Ständige Recodierung für die Ausgabe nötig. Der User soll schließlich überall den internen Code sehen und nicht den des Anbieters. Gleichzeitig sind die Referenzen zu pflegen, da Objekte auch wegfallen und sich somit Karteileichen ergeben.

11.4. Sortieroptionen in der Übersicht

Eine Sortieroption soll die Möglichkeit bieten Objekte nach Preis, Merkmalen, Typ, Anbieter und Lage zu sortieren. Da sich die Lage jedoch schon in der Suchmaske festlegen lässt, wären die einzig sinnvollen Kriterien Preis, Merkmale, Haustyp, Anbieter. Für die beiden erstgenannten Punkte ist jedoch nur eine Sortierung nach Quantität sinnvoll, denn auch die Einschränkung nach Merkmalen ist in der Suchmaske möglich. Quantitativ wäre z.B. max. Belegung, Anzahl Zimmer, Preis.

11.5. Objektanzahl in der Suchmaske

Im Zusammenhang mit Erweiterungen am Buchungssystem erscheint es sinnvoll die Suchmaske mit einem Hinweisfeld zu versehen. Es soll die Anzahl an Ergebnissen anzeigen, welche zu den gewählten Kriterien verfügbar sind. So lässt sich eine Suche ohne Ergebnisse vermeiden.

11.6. Verändertes Verfahren der Datenausgabe

Statt jegliche Datenbankabfragen und den Bezug der Bilder während des Aufbaues der Übersicht durchzuführen, sollten alle Daten im Vorfeld gesammelt werden. So lange Abfragen im Gange sind, könnte ein Hinweis mit der Bitte um Geduld eingeblendet werden. Im Hintergrund landen alle relevanten Daten in einem Zwischenspeicher und lassen sich in Echtzeit beim Aufbau der Übersicht anzeigen.

11.7. Buchungsarten

Nicht jedes Objekt kann direkt gebucht werden. Einige Objekte, im besonderen die Privateinträge des bestehenden Ferienhaus Systems, lassen sich nur durch direktes in Kontakt treten mit dem Vermieter buchen. Der Auftraggeber hätte auch hier lieber eine automatisierte Variante. Die Entscheidung darüber liegt letztendlich bei den Anbietern der Häuser. Für sie bedeutet ein automatisierter Vorgang im Endeffekt mehr Arbeit, da sie ständig auf Buchungen prüfen und ihre Belegungsdaten auf dem neuesten Stand halten müssen. Daraus ergibt sich der Kompromiss zweigleisig zu fahren, also die Option zu lassen weiterhin auf Anfragen zu setzen oder eine automatisiertes System anzunehmen. Sollte eine Automatisierung eingeführt werden, erfordert dies Erweiterungen für die Belegungsübersicht und Buchung.

11.8. Sicherheit

Da sich während des Buchungsvorganges vertrauliche Daten ansammeln ist deren Übertragung an die Anbieter in verschlüsselter Form zu realisieren. Beide Übertragungsarten können dies zum gewählten Zeitpunkt jedoch nicht. Hier ist abzuklären, ob solche Verfahren schon bereitgestellt werden und wie diese zu bedienen sind.

11.9. Reseller Option

In den anfänglichen Planungen ist es vorgesehen den Datenpool interessierten Kunden zugänglich zu machen. Sie können den Datenbestand auf ihren Seiten nutzen und erhalten Provisionen für jedes vermittelte Objekt. Ein weiterer Punkt war die Realisierung eines Werbesystems. Interessenten könnten dann Werbung für das Portal auf ihren Seiten schalten. So ließe sich die Bekanntheit des eigenen Portals steigern.

12. Resümee

12.1. Die Aufgabe

Während der Konkretisierung des Themas kamen folgende Aufgaben in den Katalog.

- Aufbau eines Datenpools
- Einspeisung von Daten mehrerer Anbieter
- Realisierung eines Frontend für den Nutzer
- Zugriff auf den Datenpool und Bereitstellung der Daten auf den Seiten von Kunden der Firma
- Entwurf eines Administrationsbereiches für Werbekunden
- Bannersystem für Werbezwecke

Während der Erstellung des Konzeptpapiers zeichnete sich ab, dass nicht alle Punkte innerhalb des gesetzten Zeitrahmen zu bewältigen sind. Da die Priorität auf den ersten drei Punkten lag wurden alle nachfolgenden Punkte zurück gestellt. Dennoch stellte sich auch die Bewältigung der verbliebenen Aufgaben als Zeitintensiv heraus.

12.2. Ergebnisse

Im ersten Schritt musste die Datenbank angelegt werden, da sie eine Voraussetzung für die Erfassung der Daten darstellt. Die Erstellung der Datenstrukturen war ein komplexer Vorgang. Es konnte nicht sofort eine Struktur entworfen werden die allen Belangen gerecht wird. Zum einen stellte einer der Anbieter bei Arbeitsbeginn seine eigenen Strukturen gerade um, zum anderen kam im Verlauf der Entwicklung ein weiterer Anbieter hinzu. Einige Aspekte kristallisierten sich erst während der Umsetzung heraus. Beispiele hierfür sind die Übertragung von Informationen in der Buchungsmaske oder Erweiterungen um neue Merkmalsangaben in der Datenbank. Die Abgrenzung einzelner Tabellen nach Aufgabenbereich und Typ wurde auch erst später durchgeführt. So gesehen wuchs die Datenbank mit Voranschreiten der Arbeiten mit und wurde an die Bedürfnisse angepasst. Die aktuelle Struktur ist für die momentanen Aufgaben geeignet. Dennoch wird es auch in Zukunft Änderungen am System geben, da Datensätze weiterer Anbieter integriert und einige Punkte verbessert werden sollen. Es muss z.B. der Fall berücksichtigt werden, dass Anbieter neue Informationen in ihre Daten einfügen.

Die Idee, sämtliche gelieferten Daten als CSV Datei neu zu schreiben, ergab sich aus der Dokumentation der MySQL Datenbank. So existieren SQL Befehle um Dateien mit einer der Tabelle entsprechenden Formatierung einzulesen. Das Konzept hat sich bewährt und findet in allen Bereichen der Datenaggregation Verwendung.

Ein Großteil der gelieferten Daten lies sich relativ einfach umwandeln. Auf Stammdaten traf dies leider nicht zu, da alle Referenzierungen angepasst werden mussten. Jeder Anbieter hat seine eigenen Strukturen, wobei die Aufteilung nach Ländern, Regionen und Orten bei allen gleich ist. Vom Entwurf einer Strategie zur Anpassung, bis zu einer nahezu fehlerfrei lauffähigen Version, verging deshalb über ein Monat! Einige Aspekte wurden nicht berücksichtigt wodurch es zu Mehrdeutigkeiten in den Zuordnungen kam. Die Fehler ließen sich jedoch erkennen und beheben. Hier bewährte sich das konsequente Testen und ständige Fehlersuche in der Funktion. Aus den Fehlern ergaben sich meist die Ursachen. Im Nachhinein muss allerdings gesagt werden, dass das System in seiner jetzigen Form nicht dem endgültigen Zustand entspricht. Es arbeitet zwar zufriedenstellend, generiert aber eine Menge redundanter Daten in der Datenbank. Zugleich gestaltet sich eine manuelle Auswertung bei Fehlern schwierig. Auch die fehlende direkte Vergleichsmöglichkeit der Daten mit anderen Anbietern begünstigt den Ruf nach einer besseren Methode. Deshalb wurde nach einer neuen Variante gesucht. Hier ergab sich während der Entwicklung ein Konzept welches mehr Effizienz verspricht. Eine praktische Realisierung steht noch aus.

Die Arbeit mit den gewonnenen Daten und der Aufbau eines Buchungssystems stellten den größten Teil der Arbeit dar. Hierbei war eine Fülle an Aspekten zu berücksichtigen. Die Entwicklung folgte dem Buchungsverlauf den ein User nimmt. So begannen die ersten Arbeiten an der Suchmaske. Sie wurde soweit ausgebaut, dass eine geografische Suche und Auswahl von Merkmalen möglich ist. Der zweite Schritt bestand aus dem Aufbau der Objektübersicht. Hier musste die Ausgabe der Objekte mit den wichtigsten Angaben geschehen. Die Anzahl dargestellter Objekte war zu begrenzen und eine Anwahl nicht dargestellter Objekte sollte möglich sein. All diese Merkmale konnten realisiert werden und arbeiten fehlerfrei. Die Detailansicht, als logischer nächster Schritt, integriert die geforderten Ausgaben und ist somit auch vollständig. Die Merkliste entspricht der Objektübersicht mit zusätzlichen Funktionen für Speicherung und Löschung. Bei den Abfragen für die Belegungsübersicht gibt es noch Nachholbedarf. Durch Änderungen am Abfrageverhalten und Datenbestand der Interhome Objekte sind Anpassungen für die Preisausgabe notwendig. Für die beiden anderen Anbieter steht keine Veränderung oder Verbesserung an. In der darauf folgenden Buchung sind nur noch Kleinigkeiten zu korrigieren. Beispiel ist hier das Setzen der Zahlungsart. Sollte aus einer alten Buchung der Punkt „Auf Rechnung“ eingestellt sein, darf dieser nicht aktiviert sein, wenn in der neuen Buchung von vornherein keine Auswahl möglich ist.

So geht es Schritt für Schritt weiter bis das Ziel der erfolgreichen Buchung erreicht ist. Ab diesem Zeitpunkt fingen die Feinarbeiten an den einzelnen Buchungsschritten an. Einige Teile wurden um Funktionen ergänzt, andere wiederum verbessert. Ebenfalls erwähnt werden muss, dass ein paar Programmteile mit der Zeit komplett verworfen und neu geschrieben wurden. Erst während der Arbeiten und dem daraus resultierenden Umfang kam die Objekt orientierte Programmierung zum Einsatz. Sie hätte von Anfang

an eingesetzt werden müssen. So war es nötig, Zeit in die Anpassung und Veränderung zahlreicher Scripte zu investieren. Ein Teil der Anpassungen betraf jedoch ähnliche Hintergründe wie sie in der Konzeption der neuen Stammdatenroutine aufgeführt sind.

In allen drei Punkten der Aufgabenstellung ließen sich die geforderten Ziele erreichen. Der notwendige Zeitumfang für die Realisierung war jedoch größer als angenommen. Das System ist funktionsfähig. Aktuell stehen noch ein paar Feinarbeiten und die Realisierung des neuen Konzeptes für die Stammdaten an. Für weitere Anbieter müssen eine angepasste Importroutine und die Kommunikation mit Web Schnittstellen des Anbieters realisiert werden.

Literaturverzeichnis

- [1] Holzner, Steven: Ajax für Dummies, 1. Aufl., Weinheim 2006
- [2] Riccardi, Greg: Datenbanksysteme mit Internet und Java-Applikationen, München 2001
- [3] Throll, Marcus/Bartosch, Oliver: Einstieg in SQL - verstehen, einsetzen, nachschlagen., Bonn 2007
- [4] Wong, Clinton: HTTP kurz und gut., 1. Auflage, Köln 2000
- [5] Kannengiesser, Matthias: Objektorientierte Programmierung mit PHP5., Poing 2007
- [6] Kannengiesser, Matthias: PHP5 / MySQL 4 - Studienausgabe., Poing 2004
- [7] Rieber, Philipp: PHP 5 & MySQL 5 - Das Kochbuch. Rezepte, Lösungen, Best Practices., 1. Aufl., Heidelberg 2007
- [8] Sklar, David/Trachtenberg, Adam: PHP 5 Kochbuch, Dt. Ausg., 2. Aufl., O'Reilly 2005
- [9] Porteneuve, Christophe: Prototype and script.aculo.us - you never knew JavaScript could do this!, Raleigh/NC 2007
- [10] Jamsa, Kris/Lalani, Suleiman/Weakley, Steve: WEB Programmierung - Einführung in die Grundlagen des Web und seiner Protokolle. Zweck, Nutzen und Anwendung von HTTP. Einführung in die Web-Seitengestaltung mit HTML und VRML., Feldkirchen 1996
- [11] Hauser, Tobias/Löwer, Ulrich M.: Web Services - die Standards., 1. Aufl., Bonn 2004
- [12] Lubkowitz, Mark: Webseiten programmieren und gestalten., 2. korrigierter Nachdruck, Bonn 2005
- [13] Menezes, Alfred J. / Oorschot, Paul C. van / Vanstone, Scott A.: Handbook of Applied Cryptography, 1996, elektronisches Angebot im PDF Format, <http://www.cacr.math.uwaterloo.ca/hac/>
- [14] Interhome AG (Hrsg.)/Interhome AG: Interhome FAQ-sheet V3., Revision 1.1, Zürich 2008c

- [15] Interhome AG (Hrsg.)/Interhome AG: Interhome Filetransfer V3., Revision 1.1, Zürich 2008a
- [16] Interhome AG (Hrsg.)/Interhome AG: Interhome Webservices V3., Revision 1.0, Zürich 2008b
- [17] Lock, T./Tui Wolters (Hrsg.)/Tui-Wolters: Spezifikation der xml-Unterkunftsdaten., Revision vom 10.4.2008, Bremen 2008c
- [18] Unicode, Inc. (Hrsg.): The Unicode Standard 4.0., Mountain View 2003, elektronisches Angebot im PDF Format, <http://www.unicode.org/versions/Unicode4.0.0/appC.pdf>
- [19] Lock, T./Tui Wolters (Hrsg.)/Tui-Wolters: XML - Booking., Revision vom 10.4.2008, Bremen 2008a
- [20] Lock, T./Tui Wolters (Hrsg.)/Tui-Wolters: XML - Erweiterung., Revision vom 10.4.2008, Bremen 2008b
- [21] Lock, T./Tui Wolters (Hrsg.)/Tui-Wolters: XML - Examples., Revision vom 21.9.2007b, Bremen 2007
- [22] Lock, T./Tui Wolters (Hrsg.)/Tui-Wolters: xmlvakanz., Revision vom 21.9.2007a, Bremen 2007
- [23] Skop, Roland/Schäfer, Mathias/Mäkitalo, Matti et al.: Die Energie des Verstehens. HTML-Dateien selbst erstellen, Version 8.1.2 vom 01.03.2007, <http://de.selfhtml.org/>, 10.04.2008
- [24] Vonhoegen, Helmut: Einstieg in XML., Elektronische Ressource, Bonn 2007
- [25] MySQL AB (Hrsg.): MySQL 5.1 Referenzhandbuch, Revision vom 12.10.2007, HTML Hilfsdatei im CHM Format, <http://dev.mysql.com/doc/>, 25.10.2007
- [26] Achour, Mehdi/Betz, Friedhelm/Dovgal, Antony et al.: PHP-Handbuch., Revision vom 13.02.2009, HTML Hilfsdatei im CHM Format, <http://de3.php.net/download-docs.php>, 16.02.2009
- [27] W3C (Hrsg.): Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation 26. November 2008., Revision vom 26.11.2008, Web-Angebot, <http://www.w3.org/TR/2008/REC-xml-20081126/>, 17.2.2009
- [28] EDM/2 (Hrsg.): Freie Online Enzyklopädie., Revision vom 11. Oktober 2008, Web-Angebot, <http://www.edm2.com>, 5.06.2009
- [29] Wikipedia (Hrsg.): Freie Online Enzyklopädie., Revision vom 2. April 2009, Web-Angebot, <http://de.wikipedia.org>, 4.04.2009

- [30] W3C (Hrsg.): Hypertext Markup Language (HTML)., Web-Angebot, <http://www.w3.org/html>, o.J.
- [31] W3C (Hrsg.): Webservices - XML Protocol Working Group., Revision vom 27.04.2007, Web Angebot, <http://www.w3.org/2000/xml/Group/>, 16.02.2009
- [32] e-domizil GmbH & Co. KG (Hrsg.): e-domizil Ferienhaus Angebote, 2008, <http://www.e-domizil.de>
- [33] Marion Kutschank Feriendomizile (Hrsg.): ferienhaus.com Ferienhaus Angebote in Südfrankreich, 2008, <http://ferienhaus.com>
- [34] INTER CHALET Ferienhaus-Gesellschaft mbH (Hrsg.): interchalet.com Ferienhaus Angebote, 2008, <http://www.interchalet.com>
- [35] sup4u GmbH (Hrsg.): tourist-online.de Ferienhaus Angebote, 2008, <http://www.tourist-online.de>
- [36] Gailly, Jean-loup/Adler, Mark: gzip, Version 1.3.3 vom 27.07.2003, <http://www.gzip.org/>, 17.04.2009
- [37] Microsoft (Hrsg.): Microsoft, 2009, <http://www.microsoft.com>
- [38] Sun Microsystems (Hrsg.): Open Office 3.1, 2009, <http://www.openoffice.org>
- [39] Open Source Initiative (Hrsg.): Verein zur Förderung von Open-Source Projekten., Revision vom 13. März 2007, Web-Angebot, <http://opensource.org>, 11.06.2009

Anhangsverzeichnis

Anhangsverzeichnis	117
A. Funktionsreferenz	120
A.1. Allgemeines	120
A.2. Funktionen der „dataload“ Klasse	120
interhome_ftpload	120
interhome_explodefile	120
stammdaten_indb	120
interhome_backupfile	121
wolters_ftpload	121
wolters_indb	121
onferienhaus_ftpload	121
onferienhaus_indb	121
geo_zuweisung	122
stammdaten_landesvergleich	122
stammdaten_regionsvergleich	122
stammdaten_ortsvergleich	122
wostammdaten_landesvergleich	123
wostammdaten_regionsvergleich	123
wostammdaten_ortsvergleich	123
onstammdaten_landesvergleich	123
onstammdaten_regionsvergleich	123
onstammdaten_ortsvergleich	123
A.3. Funktionen der „zeitfunktionen“ Klasse	123
datum_deisoun	123
datum_isode	123
days_act_month	123
days_next_month	124
thistonextmonth	124
startendactmonth	124
startendnexmonth	124
tabelle_monatskopf	124
A.4. Funktionen der „datensaetze“ Klasse	124
stammdaten_holen	124
land_holen	124
region_holen	125

ort_holen	125
haustext_holen	125
wohnungstext_holen	125
merkmale_holen	125
personenanzahl_holen	125
hausname_holen	125
quali_holen	126
hausbilder_holen	126
wohnungsbilder_holen	126
preise_holen	126
zusatzleistungen_holen	126
zusatzleistungen_holen_getrennt	127
A.5. Funktionen der „changeto_eur“ Klasse	127
eurpreis	127
A.6. Funktionen der „suchfunktionen“ Klasse	127
merkmaltoquery	127
B. Zusätzliche Scripte für Auswertung und Bearbeitung von Daten	128
B.1. Grafik Script	128
B.2. Datenbankabfrage Regionen und Orte	130
B.3. Weiterleitungsscript für Vakanzabfrage	134
B.4. Script zur Änderungen der Objektanzahl pro Seite	135
B.5. Bildausgabe aus gewähltem Vorschaubild	135
C. Aggregationsscripte für Interhome Datensätze	137
C.1. hausbilder.php	137
C.2. zimmerbilder.php	138
C.3. interhome_buchungshinweise.php	139
C.4. interhome_merkmale.php	141
C.5. interhome_preise_v2.php	144
C.6. interhome_hausprosa_v2.php	146
C.7. interhome_zimmerprosa_v2.php	148
C.8. interhome_vakanzen.php	150
C.9. interhome_zusatzleistungen.php	152
interhome_zusatzserviceentity.php	154
interhome_zusatzservicetext.php	154
interhome_zusatzservicetyp.php	155
C.10.interhome_stammdaten1_referenzierung.php	156
C.11.interhome_stammdaten2_eintragung.php	160
D. Aggregationsscripte für Wolters Datensätze	163
D.1. wolters_bilder.php	163
D.2. wolters_buchungshinweise.php	164
D.3. wolters_merkmale.php	165

D.4. wolters_preise.php	169
D.5. wolters_hausprosa.php	170
D.6. wolters_zimmerprosa.php	171
D.7. wolters_zubs.php	173
D.8. wolters_stammdaten1_referenzierung.php	174
D.9. wolters_stammdaten2_eintragung.php	179
E. Aggregationsscripte ursprüngliches auf neues System	181
E.1. Allgemeine Vorgehensweise	181
E.2. onferienhaus_stammdaten1_referenzierung	181
E.3. onferienhaus_stammdaten2_eintragung	185
E.4. hoteltocsv.php	186
F. Quelltexte der Klassen und Funktionen	195
F.1. Quelltext der Dataload Klasse	195
F.2. Quelltext der Zeitfunktionen Klasse	208
F.3. Quelltext der Geldfunktionen Klasse	210
F.4. Quelltext der Datensätze Klasse	211
F.5. Quelltext der Suchfunktionen Klasse	219
G. Quelltexte der Buchungsmaske	222
G.1. Suchmaske	222
G.2. Ergebnisübersicht	231
G.3. Ferienhaus Detailansicht	239
G.4. Ferienhaus Merkleiste	245
G.5. Belegungsübersicht	249
G.6. Vakanzausgabe allgemein	254
G.7. Preisausgabe Interhome Objekte	262
G.8. Preisausgabe Wolters Objekte	262
G.9. Preisausgabe Private Objekte	267
G.10. Genereller Variablencheck	267
G.11. Buchungsvorgang Teil 1	273
G.12. Buchungsvorgang Teil 2	291
G.13. Buchungsvorgang Teil 3	298
G.14. Datenversand Interhome	314
G.15. Datenversand Wolters	317
G.16. Überprüfungsroutine Buchungsstrecke	322

A. Funktionsreferenz

A.1. Allgemeines

Innerhalb der Datenverarbeitung und Ausgabe tauchen sich wiederholende Programmenteile auf. Um Aufwand und Platzbedarf zu minimieren, sowie spätere Modifikationen einfacher zu gestalten sind diese Programmteile innerhalb von Klassen in Funktionen gegliedert. Nachfolgend wird eine kurze Erläuterung der Funktionen gegeben.

Die zugehörigen Scripte befinden sich im Kapitel F.

A.2. Funktionen der „dataload“ Klasse

interhome_ftpload

Diese Funktion holt die Dateien vom Interhome FTP Server ab, speichert sie lokal und entpackt sie. Als Eingabe erwartet *interhome_ftpload* den Namen des lokalen Verzeichnisses, einen lokalen Dateiname und den Namen der Datei, welche vom Server kopiert werden soll. Inkludiert ist die „interhome_ftpconnect.php“. Sie stellt die nötigen Zugangsdaten bereit und baut eine Verbindung auf. Der gesamte Vorgang wird mit Statusmeldungen begleitet. Hieraus lassen sich bei Fehlern Rückschlüsse auf den betroffenen Bereich ziehen. Neben den Meldungen kommt es auch zur Rückgabe der Dateigröße, der entpackten Datei.

interhome_explodefile

Durch Aufruf dieser Funktion lässt sich eine lokale Datei in seine Zeilen zerlegen. Als Anhaltspunkt dient dabei das Wagenrücklaufszeichen „\n“. Im Vorfeld kommt es zu einer Überprüfung der Datei auf Existenz von Anführungszeichen. Da diese später als Spaltentrennung dienen sollen, müssen sie im ursprünglichen Text entfernt werden. Ist die Auftrennung nach Zeilen erfolgt ermittelt die Funktion deren Anzahl. Sie bilden zusammen mit dem Array welches alle Zeilen enthält die Rückgabewerte.

stammdaten_indb

Nach einer erfolgreichen Konvertierung der Ursprungsdaten sollen diese in die Datenbank übernommen werden. Dazu erwartet die Funktion eine Angabe zur Tabelle welche mit neuen Daten zu füllen ist. Außerdem ist die Datei zu benennen in der alle konvertierten Daten abgelegt wurden. Zusätzlich ist die Angabe der Anbieterkennung pflicht.

Im ersten Schritt löscht die Funktion alle vorhandenen Datensätze des Anbieters aus der angegebenen Tabelle. Darauf folgt die Übernahme der neuen Daten mit Hilfe der SQL Anweisung *LOAD DATA LOCAL INFILE*. Eine Übernahme findet jedoch erst statt wenn die alten Datensätze auch wirklich gelöscht werden konnten. Auch in diesem Fall dokumentieren Statusinformationen den Verlauf.

interhome_backupfile

Es wird versucht ein Backup der herunter geladenen und entpackten Datei zu erstellen. Dazu ist eine Angabe von Ursprungs-Verzeichnis, Dateiname und Backup-Verzeichnis nötig. Die Funktion überprüft die Existenz der Datei von welcher das Backup zu erzeugen ist. Ist sie vorhanden, wird nach einem bereits vorhandenen Backup gesucht und dieses im Existenzfall gelöscht. Darauf folgt der Kopiervorgang der ein neues Backup erzeugt. Im letzten Schritt wird die ursprüngliche Datei gelöscht. Auch der Ablauf dieser Funktion ist mit Statusmeldungen versehen. Sie bilden den einzigen Rückgabewert.

wolters_ftpload

Prinzipiell läuft die Funktion ähnlich wie *interhome_ftpload* ab, auch die erwarteten Eingabewerte besitzen dieselbe Charakteristik. Jedoch wird hierbei der FTP Server von Tui-Wolters angesprochen. Da die Dateien nicht als Zip, sondern gz Archiv, bereit stehen ist auch der Entpackvorgang anders. Es kommen nicht die ZIP, sondern gzip Funktionen von PHP zum Einsatz. Der Ablauf dagegen bleibt der gleiche

wolters_indb

Hierbei handelt es sich um eine Funktion, welche für die Eintragung von Wolters Datensätzen zuständig ist. Aufgrund einiger Verbesserungen wurde sie durch die allgemeinere Funktion *stammdaten_indb* abgelöst und ist nicht mehr nötig.

onferienhaus_ftpload

Kopiert die vom Exportsript erzeugten Dateien vom ursprünglichen auf den neuen Ferienhaus Server. Das Entpacken der Daten ist nicht, nötig da sie unverpackt und schon in ihrer endgültigen Form auf dem Server liegen.

onferienhaus_indb

Diese Funktion las alle Datensätze in die Datenbank, welche vom alten Ferienhaus System exportiert wurden. Als Ersatz dient auch wie schon bei *wolters_indb* geschildert, die Funktion *stamndateb_indb*

geo_zuweisung

Dient dazu Codierungen von Ländern, Regionen und Orten, wie sie Interhome benutzt, aus den jeweiligen Dateien auszulesen. Sie werden überprüft und als Array an das Script zurück geschickt, welches für die Aktualisierung von Datensätzen in der Interhome Referenztabelle zuständig ist.

stammdaten_landesvergleich

Die Funktion erhält den Namen eines Landes und versucht daraufhin aus der Referenztabelle das zugehörige Land auszuwählen. Dabei sind drei Fälle zu berücksichtigen. Es können mehrere Einträge, nur ein Eintrag oder gar kein Eintrag gefunden werden. Im ersten Anlauf findet eine Suche unter Einbeziehen von Wildcards am Anfang des Namens statt, so erhöht sich die Trefferwahrscheinlichkeit. Sollten nun mehrere Ergebnisse erscheinen beginnt die Suche von neuem, in dem Fall ohne Wildcards. Im Idealfall liegt nur ein Ergebnis vor. Darin enthalten ist auch die von Ferienhaus intern vergebene ID des Landes. Sie wird an das Script welches weitere Referenzierungsschritte durchführt zurückgegeben.

Es kann allerdings auch vorkommen, dass es noch keinen Eintrag gibt. In diesem Fall folgt ein Update auf die Referenztabelle der Länder und der Eintrag wird ergänzt. Da eine Suche von Land absteigend zum Ort stattfindet, gibt es natürlich weder Regionen noch Orte die dem neuen Land zugeordnet sind. Hier greift das Script vor und ergänzt neben dem Land auch gleich noch die Region und den Ort anhand des Datensatzes für dessen Konvertierung noch keine Referenz in der von Ferienhaus erstellten Referenztabelle des jeweiligen Anbieters enthalten ist.

Als wichtig ist zu erachten, dass das Nachtragen von Ländern nicht automatisiert erfolgen sollte. Für Regionen und Orte lässt sich immer auf eine höhere Instanz referenzieren, was für Länder nicht zutrifft. Dadurch kommt es oftmals zu fehlerhaften Eintragungen. Dementsprechend sollten an diesem Teil des Scriptes Änderungen vorgenommen werden.

stammdaten_regionsvergleich

Das Prinzip entspricht dem eines stammdaten Landesvergleichs. Der Schwerpunkt liegt allerdings auf den Regionen. Sollte die Region noch nicht in den Referenzen für Regionen vorhanden sein, wird auch sie ergänzt und der erste Ort zu dieser Region eingetragen.

stammdaten_ortsvergleich

Die Funktion für den Ortsvergleich arbeitet nach dem selben Muster. Sie trägt ebenfalls fehlende Einträge in den Referenztabellen nach. Da Orte die niedrigste Instanz darstellen, entfällt das Ergänzen weiterer Referenzen.

wostammdaten_landesvergleich

Stellt eine Funktion dar die speziell Daten des Anbieters TUI-Wolters auswertet und referenziert. Sie arbeitet nach dem selben Prinzip wie „stammdaten_landesvergleich“

wostammdaten_regionsvergleich

Dies trifft gleichfalls auf den Regionsvergleich zu. Er ist dem „stammdaten_regionsvergleich“ ähnlich.

wostammdaten_ortsvergleich

Auch hier lassen sich Parallelen zum „stammdaten_ortsvergleich“ ziehen.

onstammdaten_landesvergleich

Der Landesvergleich für alte Ferienhaus Objekt läuft nach dem selben Schema ab wie ein „stammdaten_landesvergleich“ .

onstammdaten_regionsvergleich

Nicht anders sieht es hier aus. Referenz ist der „stammdaten_regionsvergleich“.

onstammdaten_ortsvergleich

Auch der Ortsvergleich ist zum „stammdaten_ortsvergleich“ identisch.

A.3. Funktionen der „zeitfunktionen“ Klasse

datum_deisoun

Die Aufgabe dieser Funktion ist es ein Datum in der Schreibweise „TT.MM.JJJJ“ entgegen zu nehmen und daraus einen Unix Zeitstempel sowie die Schreibweise der Form „YYYY-MM-DD“ zu generieren. Die Funktion gibt beide Werte zurück.

datum_isode

Erwartet wird von der Funktion eine Datumsangabe in der Form „YYYY-MM-DD“. Die Form „TT.MM.JJJJ“ wie sie im deutschen Sprachgebrauch üblich ist, bildet den Rückgabewert.

days_act_month

Ermittelt aus einer Datumsangabe der Form „TT.MM.JJJJ“ die Anzahl an Tagen des Monats in dem die Datumsangabe liegt.

days_next_month

Soll ebenfalls die Anzahl aller Tage eines Monats bestimmen. Allerdings handelt es sich um den Folgemonat des gewählten Eingabedatums. Liegt als Angabe z.B. der „25.03.2009“ vor, ermittelt die Funktion die Anzahl an Tagen vom April. Dabei wird auch der mögliche Jahreswechsel berücksichtigt.

thistonextmonth

Bei dieser Funktion geht es lediglich darum einen Monat weiter zu zählen. Gleichzeitig findet die Schreibweise Berücksichtigung was den wichtigeren Punkt darstellt. Denn eine Monatsangabe ist nur dann korrekt, wenn sie immer aus 2 Stellen besteht. Also ist bei einstelligen Monaten immer eine „0“ zu ergänzen. Und genau dies tut die Funktion.

startendactmonth

Ermittelt den ersten und letzten Tag und gibt beide Daten in der Form „YYYY-MM-DD“ zurück.

startendnexmonth

Macht vom Prinzip her das gleiche wie „startendactmonth“ muss jedoch auch hier wieder den Jahreswechsel berücksichtigen.

tabelle_monatskopf

Die Funktion erwartet ein Datum als Array. Dabei müssen die Monate immer an der zweiten Stelle und Jahre an dritter Stelle im Array stehen. Anhand des Monats wird ein String gebildet, der den Namen des Monats und das Jahr des gegebenen Datums enthält.

A.4. Funktionen der „datensaetze“ Klasse

stammdaten_holen

Die Funktion erwartet Objekt- und Kategoriecode. Mit deren Hilfe lassen sich aus der Stammdaten Tabelle Landes-, Regions- und Ortsnamen für das Objekt ermitteln.

land_holen

Wird der Funktion eine Landes-ID übergeben, fragt sie aus der Referenztabelle der Länder den zugehörigen Landesnamen ab.

region_holen

Fragt statt Ländern Regionen anhand der Regions-ID aus der Referenztabelle für Regionen ab.

ort_holen

Der Ablauf ist dem der beiden vorherigen Funktionen ähnlich. Statt Ländern und Regionen wird anhand einer ID der Ortsname ermittelt.

haustext_holen

Fragt mit Hilfe des Referenzcodes eines Objektes die Beschreibung aus der Datenbank ab. In der Rückgabe sind sowohl eine verkürzte Fassung für die Übersicht als auch eine ausführliche Fassung enthalten.

wohnungstext_holen

Die Funktion benötigt neben dem Referenzcode auch den Kategoriecode des Objektes um die Beschreibungstexte der Wohnung abzurufen. Auch in diesem Fall beinhaltet die Rückgabe eine gekürzte und die komplette Fassung.

merkmale_holen

Mit Hilfe von Objekt- und Kategoriecode findet eine Anfrage an die Datenbank statt. Die Übernahme erfolgt nicht wie bei den Beschreibungstexten 1:1. Innerhalb der Funktion werden vier Arrays angelegt welche immer eine bestimmte Kategorie beschreiben. Eine ganze Reihe an Bedingungen sorgt für die nötige Auswertung der Merkmale. Im Abschnitt über die Datenbank wurde erläutert, dass Merkmale als „0“ bzw „1“ codiert sind. Aus dieser Codierung entstehen nun wieder die für den User verständlichen Merkmale. Angaben zu Anzahl und Entfernung werden mit den Begriffen ergänzt für die diese Angabe steht. Die Funktion gibt jedes Array für sich zurück, so dass im Script welches die Funktion abrufen nun vier eigenständige Arrays für die weitere Auswertung bereit stehen.

personenanzahl_holen

Ermittelt anhand von Referenz- und Kategoriecode eines Objektes die maximale Anzahl an erlaubten Erwachsenen und Kindern, welche dort wohnen dürfen. Eine Rückgabe der beiden Werte erfolgt getrennt.

hausname_holen

Manchen Häusern ist ein Name wie beispielsweise „Gasthaus Rosenhof“ zugeordnet. Er befindet sich in der Tabelle für Merkmale. Unter Angabe von Referenz- und Kategorieco-

de lässt er sich aus ihr gewinnen. Ist kein Name vorhanden, wird nur eine mit Leerzeichen gefüllte Variable zurückgegeben.

quali_holen

Interhome hat für seine Objekte sechs Qualitätskategorien und eine allgemeine Bewertung vorgesehen. Dieses System ist in die Merkmalstabelle eingeflossen. Aus der Datenbank werden alle sieben Felder gewonnen und bilden immer in der selben Reihenfolge als Array den Rückgabewert. Sollte es keine Bewertungen geben, wird ein Array erzeugt dessen sieben Felder alle mit „0“ belegt sind.

hausbilder_holen

Die Funktion erwartet nur den Referenzcode des Objektes. Er ermöglicht es, aus der Tabelle in welcher die Links zu den Hausbildern stehen, alle entsprechenden Einträge auszugeben. Alle Einträge landen der Reihe nach in einem Array. Dies gilt auch für die Nummerierung der Bilder. Im zweiten Teil soll per Zufallsgenerator ein Bild für die Vorschau ausgewählt und dessen Link zurückgegeben werden. Bei Absprachen mit den Verantwortlichen wurde jedoch klar, dass dieses System für die Kunden ungünstig ist. Ein User erkennt ein bereits gewähltes Haus vielleicht nur an dessen Bild in der Übersicht wieder. Wechseln nun die Bilder ständig, erkennt er sein Objekt nicht mehr. So wurde die Variante über das Zufallsbild aufgegeben. Statt dessen fällt die Wahl jetzt auf das erste Bild, welches für das Objekt in der Tabelle auftaucht.

wohnungsbilder_holen

Wie zu erwarten ist für eine Abfrage zusätzlich zum Referenz- auch der Kategoriecode des Objektes nötig. Neben der Abfrage aller Bildlinks und der Nummerierung der Bilder findet die zufällige Auswahl eines Vorschaubildes noch seine Anwendung. Das Vorschaubild steht in der Detailansicht über den Thumbnails.

preise_holen

Selektiert alle Preise für das Objekt aus der Datenbank. Die Abfrage ist an Bedingungen geknüpft. So muss ebenfalls eine Aufenthaltsdauer als auch das Anreisedatum gegeben sein. Die Rückgabe bildet einen String in dem Anreisedatum, Aufenthaltsdauer und der diesen Kriterien entsprechende Preis enthalten sind.

zusatzleistungen_holen

Um Zusatzleistungen abzufragen benötigt die Funktion neben Objekt- und Kategoriecode auch eine Anbieterkennung. Aus der Abfrage ergeben sich Leistung, Preis, Währung sowie Serviceart und Berechnung. Die Ergebnisse werden in ihrer gegebenen Reihenfolge in ein Array übernommen. Es bildet gleichzeitig den Rückgabewert.

zusatzleistungen_holen_getrennt

Die Funktion erwartet zwar die selben Eingabeparameter, trennt jedoch nach Berechnungsarten und speichert sie in separaten Arrays ab. Eine Trennung ist z.B. für die Ausgabe in der Buchungsmaske nötig, da freiwillige Zusatzleistungen anders behandelt werden als Pflichtleistungen.

A.5. Funktionen der „changeto_eur“ Klasse

eurpreis

Liegen Preise nicht in Euro vor rechnet die Funktion den Preis in Euro um. Dazu werden Wechselkurse aus der Datenbank abgefragt und daraus der Euro Preis berechnet. Das Ergebnis wird gerundet und auf zwei Nachkommastellen gekürzt.

A.6. Funktionen der „suchfunktionen“ Klasse

merkmaltoquery

Ihre Aufgabe ist, es aus den Merkmalen die in der Session vorliegen einen SQL konformen Abfragestring zu erzeugen. Jeder Eintrag wird auf seinen Wert überprüft. Sollte er auf „1“ stehen schreibt die Funktion die entsprechende SQL Anweisung in ein Array. Andernfalls bleibt das Feld im Array leer. Für Entfernungsangaben gilt, dass ein Wertebereich abgefragt wird, der zwischen „1“ und dem definierten Wert liegt. Mittels *implode* Befehl erfolgt am Ende das Zusammenführen der Arrayfelder zu einem String welcher zurückgegeben wird.

B. Zusätzliche Scripte für Auswertung und Bearbeitung von Daten

B.1. Grafik Script

Das Grafik Script nimmt als Quelle ein Bild auf das verlinkt wird. Es bestimmt im ersten Schritt mit Hilfe der PHP Funktion *getimagesize* seine Größe. Neben dem Bildlink erhält das Script eine Größenangabe für Länge und Seite. Die Angabe darf eine Maximalgröße von 512 Pixeln nicht überschreiten. Kommt sie darüber wird der Wert automatisch auf 512 reduziert. Der zweite Schritt besteht darin herauszufinden ob das Bild kleiner oder größer als die Vorgaben ist. Dabei wird erst die eine Seite, danach die andere Seite verglichen. Sollte eine der beiden Seiten des Originals größer als die Vorgabe sein muss das Bild verkleinert werden. Ist keine der beiden Seiten größer steht eine Vergrößerung an. Was im dritten Schritt passiert wird durch den vorherigen Schritt bestimmt. Kommt es zu einer Verkleinerung ist zunächst einmal die Berechnung eines Umrechnungsfaktors nötig. Dabei erfolgt die Berechnung separat über beide Seitenlängen. Sind sie nicht gerade gleich lang führt die Berechnung zu unterschiedlichen Ergebnissen. Interessant ist nur das Ergebnis mit dem geringeren Wert, denn er stellt den Umrechnungsfaktor dar, für den die längere Seite im ursprünglichen Bild das neue Bild komplett ausfüllt. Durch einen höheren Wert würde die längere Seite überstehen, ein kleinerer Wert führt zu einem Rahmen der auf allen Seiten des neuen Bildes auftaucht. Dieser „Verkleinerungsfaktor“ erlaubt es die neuen Seitenlängen für das Ursprungsbild zu berechnen und behält das Seitenverhältnis bei. Im Anschluss folgt die Berechnung der Rahmen. Nachkommastellen lassen sich dabei durch die PHP Funktion *floor* vermeiden. Sie rundet die Werte ab. Nun stehen alle Werte bereit um das neue Bild aufzubauen. Dazu kommen einige Funktionen der GD-Library zum Einsatz um z.B. ein neues Bild zu erstellen, die Hintergrundfarbe zu wählen und das Ursprüngliche Bild im neuen Format hinein zu kopieren.

Die Vergrößerung eines Bildes ist weniger kompliziert. Es müssen lediglich die Ränder berechnet werden. Der Aufbau des neuen Bildes ist gleich.

Listing B.1: Bildbearbeitung

```
1 $bild = $_GET['pic'];
2 $bildgroesse_source = getimagesize($bild);
3 if (isset($_GET['x']) AND is_numeric($_GET['x']) AND $_GET['x'] < 512) {
4     $size_x_orig = $_GET['x'];
5 }
6 else {
7     $size_x_orig = 512;
8 }
```

```

9  if(isset($_GET['y']) AND is_numeric($_GET['y']) AND $_GET['y'] < 512) {
10    $size_y_orig = $_GET['y'];
11  }
12  else {
13    $size_y_orig = 512;
14  }
15
16  $src_x = $bildgroesse_source[0];
17  $src_y = $bildgroesse_source[1];
18  // $ar_src = $src_x/$src_y;
19
20  /**
21   * AR > 1 -> längere Seite auf X-Achse,
22   * AR < 1 -> längere Seite auf Y-Achse,
23   * AR = 1 -> Seiten gleich lang
24   * Dest AR = 1.333 als Referenz für Fallunterscheidung
25   * (AR 1.333 ergibt sich bei 320x240)
26   */
27
28  //Checken ob Bild größer oder kleiner als vorgebe ist,
29  //beide Seiten, längere entscheidet
30  $xmax = 0;
31  $ymax = 0;
32  if($src_x > $size_x_orig) {
33    $xmax = 1;
34  }
35  if($src_y > $size_y_orig) {
36    $ymax = 1;
37  }
38
39  //Wenn einer der beiden Parameter auf 1 ist,
40  //Bild im gesamten verkleinern
41  $faktor = 0;
42  if ($xmax == 1 OR $ymax == 1) {
43    //Berechnung Faktoren für beide Seiten
44    $faktorx = $size_x_orig/$src_x;
45    $faktory = $size_y_orig/$src_y;
46    //Entnahme des Faktors, der kleiner ist
47    if ($faktorx < $faktory) {
48      $faktor = $faktorx;
49    }
50    elseif($faktorx > $faktory) {
51      $faktor = $faktory;
52    }
53    //Berechnung neuer Seitenlängen nach ermitteltem Faktor
54    $sizenewy = $src_y*$faktor;
55    $sizenewx = $src_x*$faktor;
56
57    //berechnung Ränder beider Seiten, durch 2,
58    //da Randaufteilung jeweils auf beiden Seiten
59    $rand_x = floor(($size_x_orig-$sizenewx)/2);
60    $rand_y = floor(($size_y_orig-$sizenewy)/2);
61    //Korrektur Rundungsfehler
62    if ($rand_x < 0) {
63      $rand_x = 0;
64    }
65    if ($rand_y < 0) {
66      $rand_y = 0;
67    }
68
69    $destX = $rand_x;
70    $destY = $rand_y;
71    $destW = floor($sizenewx);
72    $destH = floor($sizenewy);
73
74

```

```

75 //Vorlagebild in Variable
76 $im = ImageCreateFromJPEG($bild);
77 //Neue "Bühne" für neues Bild
78 $im2 = @imagecreatetruecolor($size_x_orig, $size_y_orig)
79       or die("Cannot initialize new GD image stream");
80 $bck_color = imagecolorallocate($im2, 255, 248, 238);
81 ImageFill($im2, 0,0, $bck_color);
82 ImageCopyResampled($im2, $im, $destX,$destY,0,0,$destW,$destH,$src_x,$src_y);
83 /**
84  * $im -> Das zu übertragene Bild
85  * $im2 -> Neue Bühne auf die übertragen wird,
86  * grÖße bestimmt durch die Parameter in den Klammern
87  * destX/destY -> Abstand der Source zum linken
88  * und oberen Rand in der Destination
89  * srcX/srcY -> Beschneidungsbereich source ab PosX
90  * und PosY (nicht erwünscht, also 0,0)
91  * destW/destH -> BildgrÖße Width/Height der Source auf dem neuen Bild
92  * srcW/srcH -> Beschneidungsbereich source ab posX bis
93  * Width und ab PosY bis Height (Da nicht beschneiden also
94  * genauso groß wie Source)
95  * destimg-resimg-destX-destY-srcX-srcY-destW-destH-srcW-srcH
96  */
97 ImageJPEG($im2);
98 }
99 else {
100 //berechnung Ränder beider Seiten, durch 2, da
101 //Randaufteilung jeweils auf beiden Seiten
102 $rand_x = floor(($size_x_orig-$src_x)/2);
103 $rand_y = floor(($size_y_orig-$src_y)/2);
104 //Korrektur Rundungsfehler
105 if ($rand_x < 0) {
106     $rand_x = 0;
107 }
108 if ($rand_y < 0) {
109     $rand_y = 0;
110 }
111 $destX = $rand_x;
112 $destY = $rand_y;
113 $destW = floor($src_x);
114 $destH = floor($src_y);
115
116 //Vorlagebild in Variable
117 $im = ImageCreateFromJPEG($bild);
118 //Neue "Bühne" für neues Bild
119 $im2 = @imagecreatetruecolor($size_x_orig, $size_y_orig)
120       or die("Cannot initialize new GD image stream");
121 $bck_color = imagecolorallocate($im2, 255, 248, 238);
122 ImageFill($im2, 0,0, $bck_color);
123 ImageCopyResampled($im2, $im, $destX,$destY,0,0,$destW,$destH,$src_x,$src_y);
124 ImageJPEG($im2);
125 }

```

B.2. Datenbankabfrage Regionen und Orte

In der Suchmaske hat der User die Möglichkeit nach der Auswahl eines Landes zugehörige Regionen und deren Orte zu wählen. Durch Wahl eines Landes aktualisiert sich das Pull-Down Menü der Regionen. Die Daten werden über ein Script bereit gestellt welches bis zu drei Werte entgegen nimmt mit denen eine Datenbankabfrage durchgeführt wird. Die Werte enthalten entweder ein Land zu dem alle Regionen gesucht werden müssen oder eine Region für die alle Orte zu ermitteln sind. Der dritte Parameter ist für die

Volltextsuche reserviert.

Wählt der User ein Land erhält das Script dessen ID und beginnt eine Suche nach allen Regionen die mit dieser ID übereinstimmen. Zu jeder passenden Region wird eine weitere Abfrage in der Datenbank durchgeführt. Sie hat die Aufgabe die Anzahl der Objekte jeder Region zu ermitteln. Gleichzeitig ist sie mit der Bedingung verknüpft alle Regionen auszublenden in denen es keine Objekte gibt. Die gültigen Ergebnisse werden als Auswahl innerhalb des Pull-Down Menüs erstellt.

Eine Auflistung aller Orte der gewählten Region läuft nach dem selben Muster ab, nur dass eben nach Orten gesucht wird.

Eine Volltextsuche löst Abfragen von Ländern, Regionen und Orten aus. Alle drei Abfragen laufen nach dem gleichen Muster ab wie es für die Regionen bereits erläutert wurde. Der Unterschied besteht darin, dass die Ergebnisse nicht in einem Menü, sondern als Array zwischengespeichert werden. Im zweiten Teil folgt eine Überprüfung auf Existenz von Ergebnissen in den drei Arrays. Dabei bildet das Script aus den Einträgen ein Auswahlmenü, absteigend sortiert nach Ländern, Regionen und Orten.

Listing B.2: Abfrage Regionen Orte

```

1 <?php
2 $pfad = getenv("DOCUMENT_ROOT");
3 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
4 include $datenbankzugriff;
5
6 if($_GET['L1'] != '' && $_GET['L2'] == 'NULL') {
7     $land = $_GET['L1']; //Nur dadurch auch Orte auswählbar
8     ?>
9     <table>
10        <tr>
11            <td class="geotd1">Region:</td>
12            <td>
13                <select class="geoselect" name="L2" size="1"
14                    onchange="UpdateDiv2(this.value,'<?php_echo_<$_GET['L1']>');">
15                    <option value="kA">Region</option>
16                    <?php
17                        $countrycode = "WHERE_LandID_=$_GET['L1']" . $_GET['L1'];
18                        //SQL Abfrage
19                        $query = "SELECT_*_FROM_menu_region_<$_GET['L1']>_ORDER_BY_Region";
20                        $result = mysql_query($query) or die("Anfrage_gescheitert:" . mysql_error());
21                        //Ergebnis
22                        while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
23                            $regionscode = $line["IDRegion"];
24                            $regionsname = utf8_encode($line["Region"]);
25                            $query_region_objekte = "SELECT_Region_FROM
26 .....obj_stammdaten_WHERE_Region_=$_GET['L1']>";
27                            $result_region_objekte = mysql_query($query_region_objekte)
28                            or die("Anfrage_gescheitert:" . mysql_error());
29                            $objekte_region_anzahl = mysql_num_rows($result_region_objekte);
30                            if ($objekte_region_anzahl > 0) {
31                                echo '<option value="' . $regionscode . '">' . $regionsname .
32                                ' (' . $objekte_region_anzahl . ')</option>';
33                            }
34                        }
35                        //Freigabe Resultset
36                        mysql_free_result($result);
37                    ?>
38                </select>
39            </td>
40        </tr>
41    </table>

```

```

42  <?php
43  // Verbindung schliessen
44  mysql_close($db-open);
45  }
46
47  if($_GET['L2'] != 'NULL' && $_GET['L3'] == 'NULL') {
48      $countrycode = "WHERE_LandID=$_". $_GET['L1']. " '$_'";
49      $regioncode = "AND_RegionID=$_". $_GET['L2']. " '$_'";
50      $pfad = getenv("DOCUMENT_ROOT");
51      ?>
52      <table>
53          <tr>
54              <td class="geotd1">Ort:</td>
55              <td>
56                  <select class="geoselect" name="L3" size="1">
57                      <option value="kA">Ort</option>
58                      <?php
59                      //SQL Abfrage
60                      $query = "SELECT_*_FROM_menu_ort_$countrycode_$regioncode_ORDER_BY_Ort";
61                      $result = mysql_query($query) or die("Anfrage_gescheitert:" . mysql_error());
62
63                      //Ergebnis
64                      while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
65                          $ortscode = $line["IDOrt"]; $ortsname = utf8_encode($line["Ort"]);
66                          $query_ort_objekte = "SELECT_Ort_FROM_obj_stammdaten
67  -----WHERE_Ort=$_' $ortscode ";
68                          $result_ort_objekte = mysql_query($query_ort_objekte)
69                          or die("Anfrage_gescheitert:" . mysql_error());
70                          $objekte_ort_anzahl = mysql_num_rows($result_ort_objekte);
71                          if ($objekte_ort_anzahl > 0) {
72                              echo '<option_value="'. $ortscode. '">'. $ortsname.
73                                  ' ('. $objekte_ort_anzahl. ')</option>';
74                          }
75                      }
76                      ?>
77                  </select>
78              </td>
79          </tr>
80      </table>
81      <?php
82      //Freigabe Resultset
83      mysql_free_result($result);
84      mysql_close($db-open);
85  }
86
87  if($_GET['L2'] == 'SUCHE' && $_GET['L1'] == 'NULL') {
88      $wunschziel = utf8_decode($_GET['L3']);
89      if (strlen($wunschziel) > 2) {
90          $k = 0;
91          $i = 0;
92          $wunsch_land = NULL;
93          $query_land = "SELECT_*_FROM_menu_land_WHERE_Land_LIKE_ '%$wunschziel%'";
94          $abfrage_land = mysql_query($query_land);
95          if (mysql_num_rows($abfrage_land) > 0) {
96              while($line_abfr_land = mysql_fetch_array($abfrage_land, MYSQL_ASSOC)) {
97                  $landescode = $line_abfr_land["IDLand"];
98                  $landesname = utf8_encode($line_abfr_land["Land"]);
99                  $query_land_objekte = "SELECT_Land_FROM_obj_stammdaten
100 -----WHERE_Land=$_' $landescode ";
101                  $result_land_objekte = mysql_query($query_land_objekte)
102                  or die("Anfrage_gescheitert:" . mysql_error());
103                  $objekte_land_anzahl = mysql_num_rows($result_land_objekte);
104                  if ($objekte_land_anzahl > 0) {
105                      $wunsch_land[$i] = array(utf8_encode($line_abfr_land['Land']),
106                          utf8_encode($line_abfr_land['Land']). '&nbsp;(' .
107                          $objekte_land_anzahl. '&nbsp;Objekte)', $line_abfr_land['IDLand']);

```

```

108         $i++;
109     }
110 }
111 }
112 $i = 0;
113 $wunsch_region = NULL;
114 $query_region = "SELECT_*_FROM_menu_region_WHERE_Region_LIKE_'" . $wunschziel . "'";
115 $abfrage_region = mysql_query($query_region);
116 if (mysql_num_rows($abfrage_region) > 0) {
117     while($line_abfr_region = mysql_fetch_array($abfrage_region, MYSQL_ASSOC)) {
118         $regionscode = $line_abfr_region["IDRegion"];
119         $regionsname = utf8_encode($line_abfr_region["Region"]);
120         $query_region_objekte = "SELECT_Region_FROM_obj_stammdaten
121 .....WHERE_Region_=_" . $regionscode . "'";
122         $result_region_objekte = mysql_query($query_region_objekte)
123         or die("Anfrage_gescheitert:" . mysql_error());
124         $objekte_region_anzahl = mysql_num_rows($result_region_objekte);
125         if ($objekte_region_anzahl > 0) {
126             $wunsch_region[$i] = array(utf8_encode($line_abfr_region['Region']),
127             utf8_encode($line_abfr_region['Region']),
128             '&nbsp;(' . $objekte_region_anzahl . '&nbsp;Objekte)',
129             $line_abfr_region['IDRegion']);
130             $i++;
131         }
132     }
133 }
134 $i = 0;
135 $wunsch_ort = NULL;
136 $query_ort = "SELECT_*_FROM_menu_ort_WHERE_Ort_LIKE_'" . $wunschziel . "'";
137 $abfrage_ort = mysql_query($query_ort);
138 if (mysql_num_rows($abfrage_ort) > 0) {
139     while($line_abfr_ort = mysql_fetch_array($abfrage_ort, MYSQL_ASSOC)) {
140         $ortscode = $line_abfr_ort["IDOrt"];
141         $ortsname = utf8_encode($line_abfr_ort["Ort"]);
142         $query_ort_objekte = "SELECT_Ort_FROM_obj_stammdaten
143 .....WHERE_Ort_=_" . $ortscode . "'";
144         $result_ort_objekte = mysql_query($query_ort_objekte)
145         or die("Anfrage_gescheitert:" . mysql_error());
146         $objekte_ort_anzahl = mysql_num_rows($result_ort_objekte);
147         if ($objekte_ort_anzahl > 0) {
148             $wunsch_ort[$i] = array(utf8_encode($line_abfr_ort['Ort']),
149             utf8_encode($line_abfr_ort['Ort']), (' . $objekte_ort_anzahl .
150             '&nbsp;Objekte)', $line_abfr_ort['IDOrt']);
151             $i++;
152         }
153     }
154 }
155 if (is_array($wunsch_land) OR is_array($wunsch_region) OR is_array($wunsch_ort)) {
156     ?>
157     <select class="geosuche" name="direktsuche" size="7"
158     onchange="Clear(this.value);">
159     <?php
160     if (is_array($wunsch_land)) {
161         echo 'Laender<br>';
162         foreach ($wunsch_land AS $wuland) {
163             echo '<option value="' . $wuland[2] . '" . $wuland[0] . '">(Land)&nbsp;';
164             $wuland[1] . '</option><br>';
165         }
166     }
167     if (is_array($wunsch_region)) {
168         echo 'Regionen<br>';
169         foreach ($wunsch_region AS $wuregion) {
170             echo '<option value="' . $wuregion[2] . '" . $wuregion[0] .
171             '">(Region)&nbsp;';
172             $wuregion[1] . '</option><br>';
173         }
174     }
175 }

```

```

174         if (is_array($wunsch_ort)) {
175             echo 'Orte<br>';
176             foreach ($wunsch_ort AS $wuort) {
177                 echo '<option_value="'. $wuort[2]. ' '. $wuort[0].
178                     '>(Ort)&nbsp;'. $wuort[1]. '</option><br>';
179             }
180         }
181         ?>
182         </select>'
183 .....<?php
184 .....}
185 ..}
186 }
187 ?>

```

B.3. Weiterleitungsscript für Vakanzabfrage

Jeder Anbieter hat eine eigenes System was die Bereitstellung von Vakanzen und Preisen angeht. Deshalb führt ein Abruf der Daten zunächst über das Weiterleitungsscript. Es übernimmt wichtige Daten wie Referenz- und Kategoriecode eines Objektes, Aufenthaltsdauer, Anreisetag und stellt sie den nachfolgenden Scripten zur Verfügung. Sollte kein Anreisedatum gegeben sein wird dieses aus dem aktuellen Datum erstellt. Anhand der übergebenen Anbieterkennung und einer Aktionsbeschreibung lassen sich die richtigen Scripte inkludieren.

Listing B.3: Weiterleitung zur Vakanzabfrage

```

1 <?php
2 $aufenthalt_start = $_GET['date'];
3 $objektreferenz = $_GET['oref'];
4 $objektkategorie = $_GET['okat'];
5 $aufenthalt_dauer = $_GET['days'];
6 $anbieter = $_GET['anbieter'];
7 $aktion = $_GET['aktion'];
8 $sessionid = $_GET['sid'];
9
10 $pfad = getenv("DOCUMENT_ROOT");
11 $interhome_vakanz_live = $pfad.'/ferienhaus-suche/live/interhome_vakanzausgabe.php';
12 $interhome_vakanzwahl = $pfad.'/ferienhaus-suche/live/interhome_preisausgabe.php';
13 $wolters_vakanz_live = $pfad.'/ferienhaus-suche/live/wolters_vakanzausgabe.php';
14 $wolters_prices = $pfad.'/ferienhaus-suche/live/wolters_preisausgabe.php';
15 $onfh_vakanz_live = $pfad.'/ferienhaus-suche/live/onfh_vakanzausgabe.php';
16 $onfh_prices = $pfad.'/ferienhaus-suche/live/onfh_preisausgabe.php';
17 $inc_datumsklasse = $pfad.'/include/class_datumfkt.php';
18 include $inc_datumsklasse;
19
20 $datumfkt = new zeitfunktionen;
21
22 if ($aktion == 'kalender') {
23     echo '<p>Vakanzen</p>';
24     if ($aufenthalt_start == '..') {
25         $aufenthalt_start = date("d.m.Y");
26     }
27     if ($anbieter == 1) {
28         include $interhome_vakanz_live;
29     }
30     if ($anbieter == 2) {
31         include $wolters_vakanz_live;
32     }
33     if ($anbieter == 99) {

```



```

34     include $onfh_vakanz_live;
35 }
36 }
37
38 if ($aktion == 'preis') {
39     echo '<p>Preise</p>';
40     if ($anbieter == 1) {
41         include $interhome_vakanzwahl;
42     }
43     if ($anbieter == 2) {
44         include $wolters_prices;
45     }
46     if ($anbieter == 99) {
47         include $onfh_prices;
48     }
49 }
50 ?>

```

B.4. Script zur Änderungen der Objektanzahl pro Seite

Dieses Script erhält über einen XMLHttpRequest die Angabe wie viele Objekte pro Seite in der Übersicht angezeigt werden sollen. Es trägt den Wert in die Session ein und überschreibt damit die alte Angabe.

Listing B.4: Änderung Objektanzahl pro seite

```

1 <?php
2 session_start();
3 $listlaeng = $_GET['L1'];
4
5 if ($listlaeng == '10') {
6     $_SESSION['navigation'][2] = 10;
7 }
8 elseif ($listlaeng == '20') {
9     $_SESSION['navigation'][2] = 20;
10 }
11 elseif ($listlaeng == '50') {
12     $_SESSION['navigation'][2] = 50;
13 }
14 ?>

```

B.5. Bildausgabe aus gewähltem Vorschaubild

In der Detailansicht soll ein Bild aus der Gallerie in Großansicht erscheinen. Um die Seite nicht neu zu laden geschieht dies über einen XMLHttpRequest der auf das hier beschriebene Script verweist. Es übernimmt den übergebenen Bildlink und erstellt daraus ein html konformes Bildobjekt. Sollte unter dem Link kein Bild mehr existieren wird ein Standardbild ausgegeben.

Listing B.5: Änderung Galeriebild

```

1 <?php
2 $pfad = getenv("DOCUMENT_ROOT");
3 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
4 include $datenbankzugriff;
5 //Tiefe zum Stammverzeichnis ermitteln

```

```
6 $downward = "";
7 $Pfad = getenv("REQUEST_URI");
8 $Pfadteile = explode("/", $Pfad);
9 for ($i=count($Pfadteile)-1; $i>=1; $i--)
10 {
11     $downward = $downward . "../";
12 }
13
14 if ($_GET['L1'] != '') {
15     $bildlink = $_GET['L1'];
16     $bildopen = @fopen($bildlink, "r");
17     if ($bildopen) {
18         echo '<img_class="ajaxbild" _src="' . $downward .
19             'include/grafik.php?pic=' . $bildlink . '&x=320&y=240" _/>';
20     }
21     else {
22         echo 'Logo';
23         echo '<img_src="ibelogo.png" _/>';
24     }
25 }
26 }
27 ?>
```

C. Aggregationsscripte für Interhome Datensätze

C.1. hausbilder.php

Listing C.1: Aggregation von Hausbildern - Interhome

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip = 'HousePictures.csv.zip';
5 $dateiname = 'HousePictures.csv';
6
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $dataload_inc;
10
11 $strenner = '";"';
12 $strenner2 = '";';
13 $anbieter = 1;
14 $bildnummer = 1;
15 $objreferenz_alt = 0;
16 $bildtext_alt = 0;
17
18 $dataload = new dataload;
19
20 /* Daten vom FTP Server holen */
21 $dataload->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
22 $dateigroesse = $dataload->dateigroesse;
23 echo $dataload->info_ftpload;
24
25 /* Neue CSV Datei erstellen */
26 $dataload->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
27 $zeile = $dataload->zeilen;
28 $eintraege = $dataload->eintraege;
29 echo $eintraege.'_Zeilen<br>';
30 $dateiwrite = tempnam($objektverz, 'abc');
31
32 /* ----- Datei wird zum Schreiben geflusht und dann geöffnet ----- */
33 if (is_writable($dateiwrite)) {
34     echo ('Datei_'. $dateiwrite.'_ist_beschreibbar, _fahre_mit_Verarbeitung_fort...');
35 }
36 else {
37     echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden.<br>
38     _Die_weitere_Verarbeitung_wird_abgebrochen!!!';
39     mysql_close($db_open);
40     unlink($dateiwrite);
41     //Logdatei öffnen und mit Zeitstempel versehenen Eintrag einfügen
42     exit();
43 }
44 $handle_writefile = fopen($dateiwrite, "a");
45 $i = 0;
46 while ($i < $eintraege) {

```

```

47     $spaltenwerte = explode(";", $zeile[$i]);
48     $eintraege_zeile = count($spaltenwerte);
49     $i++;
50     $objreferenz = trim($spaltenwerte[0]);
51     $bildtext = trim($spaltenwerte[1]);
52     $bildlink = trim($spaltenwerte[3]);
53     if (preg_match("/\b$objreferenz\b/i", $objreferenz_alt)) {
54         $bildnummer++;
55     }
56     else { $bildnummer = 1; }
57
58     if ($objreferenz != '' OR !preg_match("/\b$bildtext\b/i", $bildtext_alt)) {
59         $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.$trenner.$bildnummer.
60         $trenner.$bildtext.$trenner.$bildlink.$trenner2."\n";
61         fwrite($handle_writefile, $eintragen);
62     }
63     $objreferenz_alt = $objreferenz;
64     $bildtext_alt = $bildtext;
65 }
66
67
68 /* einlesen Daten in die Datenbank */
69 include $datenbankzugriff;
70 $tabelle = 'obj-hauspic';
71
72 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
73 echo $dataload->info_indb;
74
75 mysql_close($db_open);
76 fclose($handle_writefile);
77 unlink($dateiwrite);
78
79 $dataload->interhome_backupfile($objektverz, $backupverz, $dateiname);
80 echo $dataload->info_backupfile;

```

C.2. zimmerbilder.php

Listing C.2: Aggregation von Wohnungsbildern - Interhome

```

1 $pfad = getenv("DOCUMENT_ROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip = 'ApptPictures.csv.zip';
5 $dateiname = 'ApptPictures.csv';
6
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $dataload_inc;
10
11 $trenner = '";"';
12 $trenner2 = '""';
13 $anbieter = 1;
14 $bildnummer = 1;
15 $bildtext_alt = 0;
16
17 /* — Daten vom FTP Server holen — */
18 $dataload = new dataload;
19 $dataload->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
20 $dateigroesse = $dataload->dateigroesse;
21 echo $dataload->info_ftpload;
22
23 /* Neue CSV Datei erstellen */
24 $dataload->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
25 $zeile = $dataload->zeilen;

```

```

26 $eintraege = $dataloader->eintraege;
27 echo $eintraege.'_Zeilen<br>';
28
29
30 /* ----- Datei wird zum Schreiben geflusht und dann geöffnet ----- */
31 $dateiwrite = tempnam($objektverz, 'abc');
32 if (is_writable($dateiwrite)) {
33     $handle_writefile = fopen($dateiwrite, "a");
34     $i = 0;
35     while ($i < $eintraege) {
36         $spaltenwerte = explode(";", $zeile[$i]);
37         $eintraege_zeile = count($spaltenwerte);
38         $i++;
39         $origobjreferenz = $spaltenwerte[0];
40         $referenznummer = explode(".", $spaltenwerte[0]);
41         $objreferenz = trim($referenznummer[0].'.'. $referenznummer[1]);
42         $kategorie = trim($referenznummer[2]);
43         $bildtext = trim($spaltenwerte[1]);
44         $saison = trim($spaltenwerte[2]);
45         $bildlink = trim($spaltenwerte[3]);
46
47         if (preg_match("/\b$origobjreferenz\b/i", $origobjreferenz_alt)) {
48             $bildnummer++;
49         }
50         else { $bildnummer = 1; }
51
52         if ($origobjreferenz != '' OR !preg_match("/\b$bildtext\b/i", $bildtext_alt)) {
53             $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.$trenner.$kategorie.
54             $trenner.$saison.$trenner.$bildnummer.$trenner.$bildtext.$trenner.
55             $bildlink.$trenner2."\n";
56             fwrite($handle_writefile, $eintragen);
57         }
58         $origobjreferenz_alt = $origobjreferenz;
59         $bildtext_alt = $bildtext;
60     }
61 }
62 else { echo 'Datei konnte zum Schreiben nicht geöffnet werden'; }
63
64 /* Einlesen in die Datenbank */
65 include $datenbankzugriff;
66 $stabelle = 'obj_wohnungspic';
67
68 $dataloader->stammdaten_indb($stabelle, $dateiwrite, $anbieter);
69 echo $dataloader->info_indb;
70
71 mysql_close($db_open);
72 fclose($handle_writefile);
73 //unlink($dateiwrite);
74
75 $dataloader->interhome_backupfile($objektverz, $backupverz, $dateiname);
76 echo $dataloader->info_backupfile;

```

C.3. interhome_buchungshinweise.php

Listing C.3: Aggregation von Buchungshinweisen - Interhome

```

1 $pfad = getenv("DOCUMENT_ROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip = 'ObjectAlert.csv.zip';
5 $dateiname = 'ObjectAlert.csv';
6 $dateizipdesc = 'ObjectAlertDescription.csv.zip';
7 $dateidesc = 'ObjectAlertDescription.csv';
8

```

```

9 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
10 $dataload_inc = $pfad.'/include/class_dataload.php';
11 include $dataload_inc;
12
13 $trenner = ',';
14 $trenner2 = ' ';
15 $anbieter = 1;
16 $bildnummer = 1;
17 $bildtext_alt = 0;
18
19 /* — Daten vom FTP Server holen — */
20 $dataload = new dataload;
21 $dataload->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
22 $dateigroesse = $dataload->dateigroesse;
23 echo $dataload->info_ftpload;
24
25 /* Daten vom FTP Server holen Description File*/
26 $dataload->interhome_ftpload($objektverz, $dateidesc, $dateizipdesc);
27 $dateigroessedesc = $dataload->dateigroesse;
28 echo $dataload->info_ftpload;
29
30 /* Hinweistexte in deutsch in ein Array schreiben */
31 $dataload->interhome_explodefile($objektverz, $dateidesc, $dateigroessedesc);
32 $zeile_hinweistext = $dataload->zeilen;
33 $eintraege_hinweistext = $dataload->eintraege;
34
35 $i_hwt = 0;
36 $i_hwtd = 0;
37 while ($i_hwt < $eintraege_hinweistext) {
38     $spaltenwerte_hinweistext = explode(";", $zeile_hinweistext[$i_hwt]);
39     $eintraege_zeile_hinweistext = count($spaltenwerte_hinweistext);
40     $sprache_hinweistext = substr($spaltenwerte_hinweistext[1], 0, 2);
41     $id_hinweistext = $spaltenwerte_hinweistext[0];
42     if (preg_match("/\b$sprache_hinweistext\b/i", 'de')) {
43         if ($id_hinweistext != '') {
44             $hinweistexte_array[$i_hwtd] = array(trim($spaltenwerte_hinweistext[0]),
45             $spaltenwerte_hinweistext[2]);
46             $i_hwtd++;
47         }
48     }
49     $i_hwt++;
50 }
51
52 /* Neue CSV Datei erstellen */
53 $dataload->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
54 $zeile = $dataload->zeilen;
55 $eintraege = $dataload->eintraege;
56 echo $eintraege.'_Zeilen<br>';
57
58 /* *Datei wird zum Schreiben geöffnet* */
59 $dateiwrite = tempnam($objektverz, 'abc');
60 if (is_writable($dateiwrite)) {
61     $handle_writefile = fopen($dateiwrite, "a");
62     $i = 0;
63     while ($i < $eintraege-1) {
64         $i_hwt = 0;
65         $spaltenwerte = explode(";", $zeile[$i]);
66         $eintraege_zeile = count($spaltenwerte);
67         $i++;
68         $referenznummer = explode(".", $spaltenwerte[0]);
69         $objreferenz = trim($referenznummer[0].'.'. $referenznummer[1]);
70         $kategorie = trim($referenznummer[2]);
71         $datumvon = trim($spaltenwerte[2]);
72         $datumbis = trim($spaltenwerte[3]);
73         $hinweis_text = trim($spaltenwerte[1]);
74         /* — Servicetexte aus dem Array holen — */

```

```

75     $eintraege_hinweistext=count($hinweistexte_array);
76     while ($i_hwt <= $eintraege_hinweistext) {
77         /* über preg und nicht (int) ==, da sonst fehler bei der Zuordnung von '00' */
78         if (preg_match("/\b$hinweis_text\b/i", $hinweistexte_array[$i_hwt][0])) {
79             $hinweis_beschreibung = trim(addslashes($hinweistexte_array[$i_hwt][1]));
80             break 1;
81         }
82         $i_hwt++;
83     }
84
85     $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.$kategorie.
86     $strenner.$hinweis_beschreibung.$strenner.$datumvon.$strenner.$datumbis.
87     $strenner2."\n";
88     fwrite($handle_writefile, $eintragen);
89
90 }
91 }
92 else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
93
94 /* Einlesen in die Datenbank */
95 include $datenbankzugriff;
96 $tabelle = 'obj-bu hinweise';
97
98 $datalog->stammdaten.indb($tabelle, $dateiwrite, $anbieter);
99 echo $datalog->info.indb;
100
101 mysql_close($db_open);
102 fclose($handle_writefile);
103 unlink($dateiwrite);
104
105 $datalog->interhome_backupfile($objektverz, $backupverz, $dateiname);
106 echo $datalog->info_backupfile;
107 $datalog->interhome_backupfile($objektverz, $backupverz, $dateidesc);
108 echo $datalog->info_backupfile;

```

C.4. interhome_merkmale.php

Listing C.4: Aggregation von Merkmalen - Interhome

```

1 $pfad = getenv("DOCUMENT_ROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip = 'Objects.csv.zip';
5 $dateiname = 'Objects.csv';
6
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $datalog_inc = $pfad.'/include/class_datalog.php';
9 include $datalog_inc;
10
11 $anbieter = 1;
12 $strenner = '";';
13 $strenner2 = '"';
14
15 $datalog = new datalog;
16
17 /* Daten vom FTP Server holen */
18 $datalog->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
19 $dateigroesse = $datalog->dateigroesse;
20 echo $datalog->info_ftpload;
21
22 /* Neue CSV Datei erstellen */
23 $datalog->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
24 $zeile = $datalog->zeilen;
25 $eintraege = $datalog->eintraege;

```

```

26 echo $eintraege.'_Zeilen<br>';
27 $dateiwrite = tempnam($objektverz, 'abc');
28
29 /* ----- Datei wird zum Schreiben geflusht und dann geöffnet ----- */
30 if (is_writable($dateiwrite)) {
31     $handle_writefile = fopen($dateiwrite, "a");
32     $i = 0;
33     while ($i < $eintraege) {
34         $spaltenwerte = explode(";", $zeile[$i]);
35         $eintraege_zeile = count($spaltenwerte);
36         $i++;
37         //echo $spaltenwerte[9]. '<br>';
38         $referenznummer = explode(".", $spaltenwerte[0]);
39         $objreferenz = $referenznummer[0].'.'. $referenznummer[1];
40         $kategorie = $referenznummer[2];
41         //echo $spaltenwerte[0]. '<br>';
42     }
43     /*
44     * Datenbeschreibung
45     *
46     * $spaltenwerte
47     * [6] = Haustyp (Haus, Reihenhaus, Appartement)
48     * [7] = Stockwerk des Objektes
49     * [8] = leer.
50     * [9] = Zuteilung obj zur Interhome Marke
51     * [10] = Art Aussenschwimmbad
52     * [11] = Art Hallenschwimmbad
53     * $spaltenwerte
54     * [12] = Öffnungsdatum Schwimmbad
55     * [13] = Schließdat. Schwimmbad
56     * [14] = Tennisplatz.
57     * [15] = Meer/See Entf. in m
58     * [16] = Skilift Entf. in m
59     * [17] = Link zu Objektbild Sommer
60     * [18] = Link zu Objektbild Winter
61     * $spaltenwerte
62     * [19] = Name des Hauses
63     * [20] = Max Personen ab 3Jahre
64     * [21] = Anz. Zimmer.
65     * [22] = Anz. Schlafzimmer
66     * [23] = Anz. Sterne
67     * [24] = Haustiere
68     * $spaltenwerte
69     * [25] = Rollstuhl geeignet
70     * [26] = Offener Kamin
71     * [27] = Bewertung Lage / Aussicht
72     * [28] = Bew. Aussenbereich
73     * [29] = Bew. Einrichtung
74     * [30] = Bew. Ruhe / Lage
75     * $spaltenwerte
76     * [31] = Bew. Küche
77     * [32] = Bew. Zufahrtsstraße
78     * [33] = leer
79     * [34] = Anreisetag
80     * [35] = Mindestaufenthalt (wenn abweichend von 7 Tage)
81     * [36] = Code Mindestaufenth
82     * $spaltenwerte
83     * [37] = PLZ des Ortes wo sich Objekt befindet
84     * [38] = Last Minute buchbar
85     * [39] = Kurzaufenthalt möglich
86     */
87     //Nochma checken ob das alles wirklich so stimmt
88     if ($spaltenwerte[11] == 'N') {
89         $spaltenwerte[11] = 0;
90     }
91     else { $spaltenwerte[11] = 1; }

```



```
92
93     if ($spaltenwerte[25] == 'N') {
94         $spaltenwerte[25] = 0;
95     }
96     else { $spaltenwerte[25] = 1; }
97
98     if ($spaltenwerte[24] == 'N') {
99         $spaltenwerte[24] = 0;
100     }
101     else { $spaltenwerte[24] = 1; }
102
103     if ($spaltenwerte[26] == 'N') {
104         $spaltenwerte[26] = 0;
105     }
106     else { $spaltenwerte[26] = 1; }
107
108     if ($spaltenwerte[10] == 'N') {
109         $spaltenwerte[10] = 0;
110     }
111     else { $spaltenwerte[10] = 1; }
112
113     if ($spaltenwerte[39] == 'B') {
114         $shortbreak = 1;
115         $flexbooking = 1;
116     }
117     elseif ($spaltenwerte[39] == 'S') {
118         $shortbreak = 1;
119         $flexbooking = 0;
120     }
121     elseif ($spaltenwerte[39] == 'F') {
122         $shortbreak = 0;
123         $flexbooking = 1;
124     }
125     else {
126         $shortbreak = 0;
127         $flexbooking = 0;
128     }
129     if($spaltenwerte[27] == '' OR $spaltenwerte[27] == 'X') {
130         $cevala = 0;
131     }
132     else {
133         $cevala = $spaltenwerte[27];
134     }
135     if($spaltenwerte[28] == '' OR $spaltenwerte[28] == 'X') {
136         $cevalb = 0;
137     }
138     else {
139         $cevalb = $spaltenwerte[28];
140     }
141     if($spaltenwerte[29] == '' OR $spaltenwerte[29] == 'X') {
142         $cevalc = 0;
143     }
144     else {
145         $cevalc = $spaltenwerte[29];
146     }
147     if($spaltenwerte[30] == '' OR $spaltenwerte[30] == 'X') {
148         $cevald = 0;
149     }
150     else {
151         $cevald = $spaltenwerte[30];
152     }
153     if($spaltenwerte[31] == '' OR $spaltenwerte[31] == 'X') {
154         $cevale = 0;
155     }
156     else {
157         $cevale = $spaltenwerte[31];
```

```

158     }
159     if($spaltenwerte[32] == '' OR $spaltenwerte[32] == 'X') {
160         $cevalf = 0;
161     }
162     else {
163         $cevalf = $spaltenwerte[32];
164     }
165     if($spaltenwerte[23] == '' OR $spaltenwerte[23] == 'X') {
166         $quali = 0;
167     }
168     else {
169         $quali = $spaltenwerte[23];
170     }
171     if($spaltenwerte[14] == 'Y') {
172         $tennis = 1;
173     }
174     else {
175         $tennis = 0;
176     }
177
178     $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.$kategorie.
179     $strenner.$spaltenwerte[19].$strenner.'0'.$strenner.$spaltenwerte[11].$strenner.
180     $spaltenwerte[25].$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.
181     $strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.$spaltenwerte[24].
182     $strenner.$spaltenwerte[26].$strenner.'0'.$strenner.$spaltenwerte[10].
183     $strenner.'0'.$strenner.$spaltenwerte[11].$strenner.$tennis.$strenner.'0'.$strenner.
184     $spaltenwerte[20].$strenner.'0'.$strenner.$spaltenwerte[21].$strenner.
185     $spaltenwerte[22].$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.
186     '0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.
187     $spaltenwerte[15].$strenner.'0'.$strenner.$spaltenwerte[16].$strenner.
188     $spaltenwerte[15].$strenner.$flexbooking.$strenner.$shortbreak.$strenner.$cevala.
189     $strenner.$cevalb.$strenner.$cevalc.$strenner.$cevald.$strenner.$cevale.$strenner.
190     $cevalf.$strenner.$quali.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.
191     $strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner2."\\n";
192     //echo $eintragen."<br>";
193     fwrite($handle_writefile, $eintragen);
194     }
195 }
196 else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
197
198 /* Einlesen in die Datenbank */
199 include $datenbankzugriff;
200 $tabelle = 'obj_merkmale';
201
202 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
203 echo $dataload->info_indb;
204
205 mysql_close($db_open);
206 fclose($handle_writefile);
207 unlink($dateiwrite);
208
209 $dataload->interhome_backupfile($objektverz, $backupverz, $dateiname);
210 echo $dataload->info_backupfile;

```

C.5. interhome_preise_v2.php

Listing C.5: Aggregation von Preisen - Interhome

```

1  /* init */
2  $pfad = getenv("DOCUMENTROOT");
3  $objektverz = $pfad.'/objektdaten/';
4  $backupverz = $pfad.'/objektdaten/backup/interhome/';
5  $dateinamezip = 'price5059.csv.zip';
6  $dateiname = 'price5059.csv';

```

```

7 $tabelle = 'obj-preise';
8 $abbruch = false;
9 $pos = 0;
10 $bytezeiger = 0;
11 $dn = 0;
12 $nutzzeilen = 0;
13 $anbieter = 1;
14 $stripbytes = 10485760;
15 $trenner = '";"';
16 $trenner2 = '""';
17
18 /* includes */
19 $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
20 $dataload_inc = $pfad.'/include/class_dataload.php';
21 include $datenbankzugriff;
22 include $dataload_inc;
23
24 /* Klassen initialisieren */
25 $dataload = new dataload;
26
27 /* Daten vom FTP Server holen */
28 $dataload->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
29 $dateigroesse = $dataload->dateigroesse;
30 echo $dataload->info_ftpload;
31
32 // Alte Datensätze des Anbieters in der Tabelle löschen
33 $leeren = "DELETE FROM $tabelle WHERE IDAnbieter=$anbieter";
34 if(!$tblloesch = mysql_query($leeren)) {
35     echo 'Datensätze konnten nicht gelöscht werden: '. mysql_error();
36     '<br>Verarbeitung wird abgebrochen!<br>';
37     mysql_close($db_open);
38     exit();
39 }
40 else {
41     echo 'Datenbank erfolgreich gelöscht. Fülle mit neuen Daten auf...<br>';
42 }
43
44 /* Datei öffnen und lesen */
45 $handle = fopen($objektverz.$dateiname, "r");
46
47 // Wiederholend 10MB lesen und in Dateien schreiben bis Dateiende
48 while ($abbruch !== true) {
49     $bytezeiger = $bytezeiger+$stripbytes;
50     // jeweils 10MB lesen
51     $contents = fread($handle, $stripbytes);
52     $zeile = explode("\n", $contents);
53     // Anzahl Datensätze ermitteln und einen Datensatz abziehen
54     // -2, da Feld 0 als 1. Feld zählt
55     $zeilenanzahl = count($zeile)-2;
56     echo 'ZEILENANZAHL FÜR DIESEN DURCHGANG: '.$zeilenanzahl.'<br>';
57     // temporäre Datei erzeugen und mit Einträgen füllen
58     $dateiwrite = tempnam($objektverz, $dn.'ih-preise-');
59     $dn = $dn+1;
60     if (is_writable($dateiwrite)) {
61         $handle_writefile = fopen($dateiwrite, "a");
62         for ($i = 0; $i <= $zeilenanzahl; $i++) {
63             $zeile2 = str_replace('","',"$", $zeile[$i]);
64             $zeile2 = trim($zeile2);
65             $spalte = explode(";", $zeile2);
66             $objektcode = explode(".", $spalte[0]);
67             $hauscode = $objektcode[0].'. '.$objektcode[1];
68             $zimmercode = $objektcode[2];
69             $nutzzeilen++;
70             $neuezeile = $trenner2.$anbieter.$trenner.$hauscode.
71             $trenner.$zimmercode.$trenner.$spalte[1].$trenner.
72             $spalte[2].$trenner.$spalte[3].$trenner.$spalte[4].$trenner2."<br>";

```

```

73     fwrite($handle_writefile, $neuezeile);
74 }
75 fclose($handle_writefile);
76 // Datei n in Tabelle übernehmen
77 $loadddbinfile = "LOAD_DATA_LOCAL_INFILE '$dateiwrite'
78 INTO TABLE $tabelle FIELDS TERMINATED BY ',' ENCLOSED BY '\"' ";
79 if (!$result = mysql_query($loadddbinfile)) {
80     echo 'Daten konnten nicht geschrieben werden: '. mysql_error();
81     '<br>Verarbeitung wird abgebrochen!<br>';
82     fclose($handle);
83     mysql_close($db_open);
84     unlink($dateiwrite);
85     exit();
86 }
87 else {
88     echo 'Daten erfolgreich eingelesen<br>';
89 }
90 }
91 else {
92     echo 'Datei konnte zum Schreiben nicht geöffnet werden<br>
93 Verarbeitung wird abgebrochen!';
94     fclose($handle);
95     mysql_close($db_open);
96     unlink($dateiwrite);
97     exit();
98 }
99
100 // Position des letzten Zeilenumbruchs bestimmen
101 // und Startpunkt für nächste Datei setzen
102 $startpunkt = 0;
103 $pos = 0;
104 while ($pos !== false) {
105     $sende = $pos;
106     $pos = strpos($contents, "\n", $startpunkt);
107     $startpunkt = $pos+1;
108 }
109 $rewind = $stripbytes-$sende+1;
110 echo 'REWINDBYTES: '. $rewind. '<br>';
111 // Zeiger Weiterstellen
112 fseek($handle, $bytezeiger-$rewind);
113 // Überprüfung Dateigröße zuletzt erzeugter Datei
114 // wenn 0, dann Schleife beenden da alle Daten ausgewertet
115 $bytes = filesize($dateiwrite);
116 if ($bytes < 1) {
117     $abbruch = true;
118 }
119 echo '<br>Bytestatus: '. $bytezeiger. '<br>';
120 chmod($dateiwrite, 0755); // oktal; richtiger Wert von mode
121 unlink($dateiwrite);
122 }
123 echo 'In die Datenbank übernommene Einträge: '. $nutzzeilen. '<br>';
124 fclose($handle);
125 unlink($objektverz.$dateiname);
126 mysql_close($db_open);

```

C.6. interhome_hausprosa_v2.php

Listing C.6: Aggregation von Hausbeschreibungen - Interhome

```

1 $pfad = getenv("DOCUMENT_ROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip = 'HousePictures.csv.zip';
5 $dateiname = 'HousePictures.csv';

```

```

6
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $datalog_inc = $pfad.'/include/class_datalog.php';
9 include $datalog_inc;
10
11 $trenner = '";"';
12 $trenner2 = '""';
13 $anbieter = 1;
14 $bildnummer = 1;
15 $objreferenz_alt = 0;
16 $bildtext_alt = 0;
17
18 $datalog = new datalog;
19
20 /* Daten vom FTP Server holen */
21 $datalog->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
22 $dateigroesse = $datalog->dateigroesse;
23 echo $datalog->info_ftpload;
24
25 /* Neue CSV Datei erstellen */
26 $datalog->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
27 $zeile = $datalog->zeilen;
28 $eintraege = $datalog->eintraege;
29 echo $eintraege.'~Zeilen<br>';
30 $dateiwrite = tempnam($objektverz, 'abc');
31
32 /* ----- Datei wird zum Schreiben geflusht und dann geöffnet ----- */
33 if (is_writable($dateiwrite)) {
34     echo ('Datei~'. $dateiwrite. '~ist beschreibbar, ~fahre mit ~Verarbeitung ~fort ... ');
35 }
36 else {
37     echo 'Datei~konnte zum ~Schreiben ~nicht ~geoeffnet ~werden.<br>
38     ~Die ~weitere ~Verarbeitung ~wird ~abgebrochen!!!';
39     mysql_close($db_open);
40     unlink($dateiwrite);
41     //Logdatei öffnen und mit Zeitstempel versehenen Eintrag einfügen
42     exit();
43 }
44 $handle_writefile = fopen($dateiwrite, "a");
45 $i = 0;
46 while ($i < $eintraege) {
47     $spaltenwerte = explode(";", $zeile[$i]);
48     $eintraege_zeile = count($spaltenwerte);
49     $i++;
50     $objreferenz = trim($spaltenwerte[0]);
51     $bildtext = trim($spaltenwerte[1]);
52     $bildlink = trim($spaltenwerte[3]);
53     if (preg_match("/\b$objreferenz\b/i", $objreferenz_alt)) {
54         $bildnummer++;
55     }
56     else { $bildnummer = 1; }
57
58     if ($objreferenz != '' OR !preg_match("/\b$objbildtext\b/i", $bildtext_alt)) {
59         $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.$trenner.$bildnummer.
60         $trenner.$bildtext.$trenner.$bildlink.$trenner2."~\n";
61         fwrite($handle_writefile, $eintragen);
62     }
63     $objreferenz_alt = $objreferenz;
64     $bildtext_alt = $bildtext;
65 }
66
67
68 /* einlesen Daten in die Datenbank */
69 include $datenbankzugriff;
70 $tabelle = 'obj-hauspic';
71

```

```

72 $datalog->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
73 echo $datalog->info_indb;
74
75 mysql_close($db_open);
76 fclose($handle_writefile);
77 unlink($dateiwrite);
78
79 $datalog->interhome_backupfile($objektverz, $backupverz, $dateiname);
80 echo $datalog->info_backupfile;

```

C.7. interhome_zimmerprosa_v2.php

Listing C.7: Aggregation von Wohnungsbeschreibungen - Interhome

```

1  /* init */
2  $pfad = getenv("DOCUMENT_ROOT");
3  $objektverz = $pfad.'/objektdaten/';
4  $backupverz = $pfad.'/objektdaten/backup/interhome/';
5  $dateinamezip = 'Doc2.csv.zip';
6  $dateiname = 'Doc2.csv';
7  $tabelle = 'obj_wohnungstext';
8  $abbruch = false;
9  $pos = 0;
10 $bytezeiger = 0;
11 $dn = 0;
12 $nutzzeilen = 0;
13 $anbieter = 1;
14 $stripbytes = 10485760;
15 $trenner = ',';
16 $trenner2 = ' ';
17
18 /* includes */
19 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
20 $datalog_inc = $pfad.'/include/class_datalog.php';
21 include $datenbankzugriff;
22 include $datalog_inc;
23
24 /* Klassen initialisieren */
25 $datalog = new datalog;
26
27 /* Daten vom FTP Server holen */
28 $datalog->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
29 $dateigroesse = $datalog->dateigroesse;
30 echo $datalog->info_ftpload;
31
32 // Alte Datensätze des Anbieters in der Tabelle löschen
33 $leeren = "DELETE FROM $tabelle WHERE IDAnbieter=$anbieter";
34 if (!$tblloesch = mysql_query($leeren)) {
35     echo 'Datensätze konnten nicht gelöscht werden: '. mysql_error();
36     '<br>Verarbeitung wird abgebrochen!<br>';
37     mysql_close($db_open);
38     exit();
39 }
40 else {
41     echo 'Datenbank erfolgreich gelöscht. Fülle mit neuen Daten auf... <br>';
42 }
43
44 /* Datei öffnen und lesen */
45 $handle = fopen($objektverz.$dateiname, "r");
46
47 // Wiederholend 10MB lesen und in Dateien schreiben bis Dateiende
48 while ($abbruch !== true) {
49     $bytezeiger = $bytezeiger+$stripbytes;
50     // jeweils 10MB lesen

```

```

51  $contents = fread ($handle, $stripbytes);
52  $zeile = explode("\n", $contents);
53  // Anzahl Datensätze ermitteln und einen Datensatz abziehen
54  // -2, da Feld 0 als 1. Feld zählt
55  $zeilenanzahl = count($zeile)-2;
56  echo 'ZEILENANZAHL_FÜR_DIESEN_DURCHGANG: '.$zeilenanzahl.'<br>';
57  // temporäre Datei erzeugen und mit Einträgen füllen
58  $dateiwrite = tempnam($objektverz, $dn.'-ih-zprosa-');
59  $dn = $dn+1;
60  if (is_writable($dateiwrite)) {
61    $handle_writefile = fopen($dateiwrite, "a");
62    for ($i = 0; $i <= $zeilenanzahl; $i++) {
63      $zeile2 = str_replace("'", " ", $zeile[$i]);
64      $zeile2 = trim($zeile2);
65      $spalte = explode(";", $zeile2);
66      $objektcode = explode(".", $spalte[0]);
67      $hauscode = $objektcode[0].'.$objektcode[1];
68      $zimmercode = $objektcode[2];
69      // Nur Einträge in Deutsch übernehmen
70      if (preg_match("/\bde\b/i", $spalte[1])) {
71        $nutzzeilen++;
72        $neuzeile = $strenner2.$anbieter.$strenner.$hauscode.
73        $strenner.$zimmercode.$strenner.$spalte[1].$strenner.
74        $spalte[2].$strenner2."<br>";
75        fwrite($handle_writefile, $neuzeile);
76      }
77    }
78    fclose($handle_writefile);
79    // Datei n in Tabelle übernehmen
80    $loaddbinfile = "LOAD_DATA_LOCAL_INFILE '$dateiwrite'
81    INTO TABLE $stabelle FIELDS TERMINATED BY ',' ENCLOSED BY '\"'";
82    if (!$result = mysql_query($loaddbinfile)) {
83      echo 'Daten konnten nicht geschrieben werden: '. mysql_error().
84      '<br>Verarbeitung wird abgebrochen!<br>';
85      fclose($handle);
86      mysql_close($db_open);
87      unlink($dateiwrite);
88      exit();
89    }
90    else {
91      echo 'Daten erfolgreich eingelesen<br>';
92    }
93  }
94  else {
95    echo 'Datei konnte zum Schreiben nicht geöffnet werden<br>
96    Verarbeitung wird abgebrochen!';
97    fclose($handle);
98    mysql_close($db_open);
99    unlink($dateiwrite);
100   exit();
101  }
102
103  // Position des letzten Zeilenumbruchs bestimmen
104  // und Startpunkt für nächste Datei setzen
105  $startpunkt = $stripbytes-100000;
106  $pos = 0;
107  while ($pos !== false) {
108    $ende = $pos;
109    $pos = strpos($contents, "\n", $startpunkt);
110    $startpunkt = $pos+1;
111  }
112  $rewind = $stripbytes-$ende+1;
113  echo 'REWINDBYTES: '.$rewind.'<br>';
114  // Zeiger Weiterstellen
115  fseek($handle, $bytezeiger-$rewind);
116  // Überprüfung Dateigrösse zuletzt erzeugter Datei

```

```

117 // wenn 0, dann Schleife beenden da alle Daten ausgewertet
118 $bytes = filesize($dateiwrite);
119 if ($bytes < 1) {
120     $abbruch = true;
121 }
122 echo '<br>Bytestatus:_' . $bytezeiger . '<br>';
123 chmod ($dateiwrite, 0755); // oktal; richtiger Wert von mode
124 unlink($dateiwrite);
125 }
126 echo 'In die Datenbank übernommene Einträge in deutsch:_' . $nutzzeilen . '<br>';
127 fclose($handle);
128 unlink($objektverz.$dateiname);
129 mysql_close($db-open);

```

C.8. interhome_vakanzen.php

Listing C.8: Aggregation von Belegungsdaten - Interhome

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad . '/objektdaten/';
3 $backupverz = $pfad . '/objektdaten/backup/interhome/';
4 $dateinamezip = 'DailyVacancy.csv.zip';
5 $dateiname = 'DailyVacancy.csv';
6
7 $datenbankzugriff = $pfad . '/include/opendb_ferienhaus.php';
8 $dataload_inc = $pfad . '/include/class_dataload.php';
9 include $dataload_inc;
10
11 $strenner = '";"';
12 $strenner2 = '""';
13 $anbieter = 1;
14 $eintragen = '';
15 $dataload = new dataload;
16
17 /* Daten vom FTP Server holen */
18 $dataload->interhome_ftpload($objektverz, $dateiname, $dateinamezip);
19 $dateigroesse = $dataload->dateigroesse;
20 echo $dataload->info_ftpload;
21
22 /* Neue CSV Datei erstellen */
23 $dataload->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
24 $zeile = $dataload->zeilen;
25 $eintraege = $dataload->eintraege;
26 echo $eintraege . 'Zeilen<br>';
27 $dateiwrite = tempnam($objektverz, 'abc');
28
29 if (is_writable($dateiwrite)) {
30     $handle_writefile = fopen($dateiwrite, "a");
31     for ($i=0; $i < $eintraege-1; $i++) {
32         $spaltenwerte = explode(";", $zeile[$i]);
33         $eintraege_zeile = count($spaltenwerte);
34         $origobjreferenz = trim($spaltenwerte[0]);
35         $referenznummer = explode(".", $spaltenwerte[0]);
36         $objreferenz = trim($referenznummer[0] . '.' . $referenznummer[1]);
37         $kategorie = trim($referenznummer[2]);
38         $startdatum = trim($spaltenwerte[1]);
39         $enddatum = trim($spaltenwerte[2]);
40         $vakanz = trim($spaltenwerte[3]);
41         $uebergabe = trim($spaltenwerte[4]);
42         $minstay = trim($spaltenwerte[5]);
43
44         $sd = explode("-", $startdatum);
45         $tag = $sd[2];
46         $monat = $sd[1];

```



```

47  $jahr = $sd[0];
48  $t_mon = date("t", mktime(0, 0, 0, $sd[1], $sd[2], $sd[0]));
49  $resttage = $t_mon-$tag+1;
50  //Endposition 1. Zeiger, +1 da 1. Tag bereits gültiges Datum aus dem String ist
51  $vakanzstring = NULL;
52  $uebergabestring = NULL;
53  $minstaystring = NULL;
54
55  /* Auffüllen aller Tage des Monats vor erstem angegebenen Tag */
56  for ($k=0; $k < $tag-1; $k++) {
57    //-1 da $tag bereits erster Tag eines Eintrages ist
58    $vakanzstring = $vakanzstring.'N';
59    $uebergabestring = $uebergabestring.'X';
60    $minstaystring = $minstaystring.'0';
61  }
62
63  $vakanzstring = $vakanzstring.substr($vakanz,0,$resttage);
64  $uebergabestring = $uebergabestring.substr($uebergabe,0,$resttage);
65  $minstaystring = $minstaystring.substr($minstay,0,$resttage);
66  $datum = $jahr.'-'. $monat.'-01';
67  //echo '----- Begin Objekt-----<br>';
68  $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.$kategorie.
69  $strenner.$datum.$strenner.$vakanzstring.$strenner.$uebergabestring.$strenner.
70  $minstaystring.$strenner2."\n";
71  fwrite($handle_writefile, $eintragen);
72  $startpos = $resttage;
73
74  for ($k=1; $k <= 12; $k++) {
75    $monat = $monat+1;
76    if ($monat == 13) {
77      $monat = 1;
78      $jahr = $jahr+1;
79    }
80    $t_mon = date("t", mktime(0, 0, 0, $monat, 1, $jahr));
81    //Achtung Substring beginnt bei Null zu zählen deshalb ist startpos
82    //automatisch Folgetag nach letztem Tag im vorherigen Feld
83    $vakanzstring = substr($vakanz,$startpos,$t_mon);
84    $uebergabestring = substr($uebergabe,$startpos,$t_mon);
85    $minstaystring = substr($minstay,$startpos,$t_mon);
86    if (strlen($monat) < 2) {
87      $monat = '0'.$monat;
88    }
89    $datum = $jahr.'-'. $monat.'-01';
90
91    $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.$kategorie.
92    $strenner.$datum.$strenner.$vakanzstring.$strenner.$uebergabestring.$strenner.
93    $minstaystring.$strenner2."\n";
94    fwrite($handle_writefile, $eintragen);
95    $startpos = $startpos+$t_mon;
96  }
97  //echo '----- Ende Objekt-----<br>';
98  //print_r($vakanzstring);
99  }
100 }
101 else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
102
103 /* einlesen Daten in die Datenbank */
104 include $datenbankzugriff;
105 $tabelle = 'obj-vakanzen';
106
107 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
108 echo $dataload->info_indb;
109
110 mysql_close($db_open);
111 fclose($handle_writefile);
112 unlink($dateiwrite);

```

```

113
114 $dataload->interhome_backupfile($objektverz, $backupverz, $dateiname);
115 echo $dataload->info_backupfile;

```

C.9. interhome_zusatzleistungen.php

Listing C.9: Aggregation von Zusatzleistungen - Interhome

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip_sd = 'ServiceDetail.csv.zip';
5 $dateiname_sd = 'ServiceDetail.csv';
6
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $dataload_inc;
10
11 $trenner = '";"';
12 $trenner2 = '""';
13 $slashed = "/";
14 $anbieter = 1;
15 $i = 0;
16 $pos = true;
17 $zeigerpos = 0;
18 $test = 0;
19
20 $dataload = new dataload;
21
22 /* Daten vom FTP Server holen */
23 $dataload->interhome_ftpload($objektverz, $dateiname_sd, $dateinamezip_sd);
24 $dateigroesse = $dataload->dateigroesse;
25 echo $dataload->info_ftpload;
26
27 /* Neue CSV Datei erstellen */
28 $dataload->interhome_explodefile($objektverz, $dateiname_sd, $dateigroesse);
29 $zeile = $dataload->zeilen;
30 $eintraege_main = $dataload->eintraege;
31 echo $eintraege.'_Zeilen<br>';
32 $dateiwrite = tempnam($objektverz, 'abc');
33
34 include 'interhome_zusatzlservicetext.php';
35 include 'interhome_zusatzlservicetyp.php';
36 include 'interhome_zusatzlserviceentity.php';
37
38 /* ----- Datei wird zum Schreiben geflusht und dann geöffnet ----- */
39 if (is_writable($dateiwrite)) {
40     $handle_writefile = fopen($dateiwrite, "w");
41     fclose($handle_writefile);
42     $handle_writefile = fopen($dateiwrite, "a");
43     while ($i < $eintraege_main) {
44         $i_servicetext = 0;
45         $i_servicetypen=0;
46         $i_serviceentities=0;
47         $spaltenwerte = explode(";", $zeile[$i]);
48         $eintraege_zeile = count($spaltenwerte);
49         $i++;
50         $referenznummer = explode(".", $spaltenwerte[0]);
51         $objreferenz = $referenznummer[0].'.'.$referenznummer[1];
52         $kategorie = $referenznummer[2];
53         $preis = rtrim($spaltenwerte[5]);
54
55 /* ----- Zusammenstellen der Zusatzleistungen für die CSV Datei ----- */
56 /**

```

```

57 * Datenbeschreibung
58 * $spaltenwerte
59 * [4] = Beschreibung der Services: (ServiceText.csv)
60 * [6] = Art des Service: (ServiceType.csv)
61 * [7] = Berechnungsart des Service: (ServiceEntity.csv)
62 */
63
64 $service_desc = $spaltenwerte[4];
65 $service_type = $spaltenwerte[6];
66 $service_entity = $spaltenwerte[7];
67
68 /*—— Servicetexte aus dem Array holen —— */
69 $eintraege_servicetext=count($servicetexte_array);
70 while ($i_servicetext <= $eintraege_servicetext) {
71     if (preg_match("/\b$service_desc\b/i",
72         $servicetexte_array[$i_servicetext][0])) {
73         $service_beschreibung = addslashes($servicetexte_array[$i_servicetext][1]);
74         break 1;
75     }
76     $i_servicetext++;
77 }
78
79 /*—— Servicetypen aus dem Array holen —— */
80 $eintraege_servicetypen=count($servicetypen_array);
81 //print_r($servicetypen_array);
82 while ($i_servicetypen <= $eintraege_servicetypen) {
83     if (preg_match("/\b$service_type\b/i",
84         $servicetypen_array[$i_servicetypen][0])) {
85         $service_typ = addslashes($servicetypen_array[$i_servicetypen][1]);
86         break 1;
87     }
88     $i_servicetypen++;
89 }
90
91 /*—— Serviceeinheiten aus dem Array holen —— */
92 $eintraege_serviceentities=count($serviceentities_array);
93 while ($i_serviceentities < $eintraege_serviceentities) {
94     if (preg_match("/\b$service_entity\b/i",
95         $serviceentities_array[$i_serviceentities][0])) {
96         $service_einheit = addslashes($serviceentities_array[$i_serviceentities][1]);
97         break 1;
98     }
99     $i_serviceentities++;
100 }
101
102 /*—— Zeilenweises in die Datei schreiben —— */
103 if ( $objreferenz != '.' AND $kategorie != '' ) {
104     $objreferenz = trim($objreferenz);
105     $kategorie = trim($kategorie);
106     $service_beschreibung = trim($service_beschreibung);
107     $preis = trim($preis);
108     $spaltenwerte[3] = trim($spaltenwerte[3]);
109     $service_typ = trim($service_typ);
110     $service_einheit = trim($service_einheit);
111     $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.$trenner.$kategorie.
112     $trenner.$service_beschreibung.$trenner.$preis.$trenner.$spaltenwerte[3].
113     $trenner.$service_typ.$trenner.$service_einheit.$trenner2."\n";
114     fwrite($handle_writefile , $eintragen);
115 }
116 }
117 }
118 else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
119
120 /* Einlesen in die Datenbank */
121 include $datenbankzugriff;
122 $tabelle = 'obj_zusatzleistungen';

```

```

123
124 $datalog->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
125 echo $datalog->info_indb;
126
127 mysql_close($db_open);
128 fclose($handle_writefile);
129 unlink($dateiwrite);
130 $datalog->interhome_backupfile($objektverz, $backupverz, $dateiname_sd);
131 echo $datalog->info_backupfile;

```

interhome_zusatzlserviceentity.php

Listing C.10: Zusatzscript - Service Entity

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip_se = 'ServiceEntity.csv.zip';
5 $dateiname_se = 'ServiceEntity.csv';
6
7 /* Daten vom FTP Server holen */
8 $datalog->interhome_ftpload($objektverz, $dateiname_se, $dateinamezip_se);
9 $dateigroesse = $datalog->dateigroesse;
10 echo $datalog->info_ftpload;
11
12 /* Serviceeinheiten zuordnen */
13 $i_serviceentity = 0;
14 $i_serviceentitydeutsch = 0;
15 $datalog->interhome_explodefile($objektverz, $dateiname_se, $dateigroesse);
16 $zeile_serviceentity = $datalog->zeilen;
17 $eintraege_serviceentity = $datalog->eintraege;
18
19 while ($i_serviceentity < $eintraege_serviceentity) {
20     $spaltenwerte_serviceentity = explode(";",
21     $zeile_serviceentity[$i_serviceentity]);
22     $eintraege_zeile_serviceentity = count($spaltenwerte_serviceentity);
23     $sprache_serviceentity = substr($spaltenwerte_serviceentity[2],0,2);
24     $id_serviceentity = $spaltenwerte_serviceentity[0];
25     if (preg_match("/\b$sprache_serviceentity\b/i", 'de')) {
26         if ($id_serviceentity != '') {
27             $serviceentities_array[$i_serviceentitydeutsch] =
28             array($spaltenwerte_serviceentity[0], $spaltenwerte_serviceentity[1]);
29             $i_serviceentitydeutsch++;
30         }
31     }
32     $i_serviceentity++;
33 }
34 unlink($objektverz.$dateiname_se);

```

interhome_zusatzlservicetext.php

Listing C.11: Zusatzscript - Service Texte

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip_st = 'ServiceText.csv.zip';
5 $dateiname_st = 'ServiceText.csv';
6
7 /* Daten vom FTP Server holen */
8 $datalog->interhome_ftpload($objektverz, $dateiname_st, $dateinamezip_st);
9 $dateigroesse = $datalog->dateigroesse;
10 echo $datalog->info_ftpload;
11

```

```

12 /* Servicetexte zuordnen */
13 $dataload->interhome_explodefile($objektverz, $dateiname_st, $dateigroesse);
14 $zeile_servicetext = $dataload->zeilen;
15 $eintraege_servicetext = $dataload->eintraege;
16
17 $i_servicetext = 0;
18 $i_servicetextdeutsch = 0;
19 while ($i_servicetext < $eintraege_servicetext) {
20     $spaltenwerte_servicetext = explode(";", $zeile_servicetext[$i_servicetext]);
21     $eintraege_zeile_servicetext = count($spaltenwerte_servicetext);
22     $sprache_servicetext = substr($spaltenwerte_servicetext[2], 0, 2);
23     $id_servicetext = $spaltenwerte_servicetext[0];
24     if (preg_match("/\b$sprache_servicetext\b/i", 'de')) {
25         if ($id_servicetext != '') {
26             $servicetexte_array[$i_servicetextdeutsch] =
27                 array($spaltenwerte_servicetext[0], $spaltenwerte_servicetext[1]);
28             //print_r($servicetexte_array[$i_servicetextdeutsch]);
29             $i_servicetextdeutsch++;
30             //echo $id_servicetext.'-'. $spaltenwerte_servicetext[1]. '<br>';
31         }
32     }
33     $i_servicetext++;
34 }
35 unlink($objektverz.$dateiname_st);

```

interhome_zusatzlservicetyp.php

Listing C.12: Zusatzscript - Service Typen

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip_tp = 'ServiceType.csv.zip';
5 $dateiname_tp = 'ServiceType.csv';
6
7 /* Daten vom FTP Server holen */
8 $dataload->interhome_ftpload($objektverz, $dateiname_tp, $dateinamezip_tp);
9 $dateigroesse = $dataload->dateigroesse;
10 echo $dataload->info_ftpload;
11
12 /* Servicetypen zuordnen */
13 $i_servicetyp = 0;
14 $i_servicetypdeutsch = 0;
15 $dataload->interhome_explodefile($objektverz, $dateiname_tp, $dateigroesse);
16 $zeile_servicetyp = $dataload->zeilen;
17 $eintraege_servicetyp = $dataload->eintraege;
18
19 while ($i_servicetyp < $eintraege_servicetyp) {
20     $spaltenwerte_servicetyp = explode(";", $zeile_servicetyp[$i_servicetyp]);
21     $eintraege_zeile_servicetyp = count($spaltenwerte_servicetyp);
22     $sprache_servicetyp = substr($spaltenwerte_servicetyp[2], 0, 2);
23     $id_servicetyp = $spaltenwerte_servicetyp[0];
24     if (preg_match("/\b$sprache_servicetyp\b/i", 'de')) {
25         if ($id_servicetyp != '') {
26             $servicetypen_array[$i_servicetypdeutsch] =
27                 array($spaltenwerte_servicetyp[0], $spaltenwerte_servicetyp[1]);
28             $i_servicetypdeutsch++;
29         }
30     }
31     $i_servicetyp++;
32 }
33 unlink($objektverz.$dateiname_tp);

```

C.10. interhome_stammdaten1_referenzierung.php

Listing C.13: Stammdaten - Referenzierung

```

1 $pfad = getenv("DOCUMENTROOT");
2 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
3 $objektverz = $pfad.'/objektdaten/';
4 $dateiname_objekte = 'Objects.csv';
5 $dateinamezip_objekte = 'Objects.csv.zip';
6 $dateiname_land = 'Countries.csv';
7 $dateinamezip_land = 'Countries.csv.zip';
8 $dateiname_region = 'Regions.csv';
9 $dateinamezip_region = 'Regions.csv.zip';
10 $dateiname_stadt = 'Places.csv';
11 $dateinamezip_stadt = 'Places.csv.zip';
12 $dateigroesse = filesize($dateiload);
13 include $datenbankzugriff;
14 $dataload_inc = $pfad.'/include/class_dataload.php';
15 include $dataload_inc;
16
17 $dataload = new dataload;
18 $datum_aktuell = getdate();
19
20 if (file_exists($objektverz.$dateiname_objekte)) {
21     $datum_datei = filemtime($objektverz.$dateiname_objekte);
22 }
23 $datum_differenz = $datum_aktuell[0] - $datum_datei;
24 if ($datum_differenz > 86400 OR !file_exists($objektverz.$dateiname_objekte)) {
25     /* Daten vom FTP Server holen */
26     $dataload->interhome_ftpload($objektverz, $dateiname_objekte,
27     $dateinamezip_objekte);
28     $dateigroesse_objekte = $dataload->dateigroesse;
29     echo $dataload->info_ftpload;
30 }
31
32 if (file_exists($objektverz.$dateiname_land)) {
33     $datum_datei = filemtime($objektverz.$dateiname_land);
34 }
35 $datum_differenz = $datum_aktuell[0] - $datum_datei;
36 if ($datum_differenz > 86400 OR !file_exists($objektverz.$dateiname_land)) {
37     /* Daten vom FTP Server holen */
38     $dataload->interhome_ftpload($objektverz, $dateiname_land, $dateinamezip_land);
39     $dateigroesse_land = $dataload->dateigroesse;
40     echo $dataload->info_ftpload;
41 }
42
43 if (file_exists($objektverz.$dateiname_region)) {
44     $datum_datei = filemtime($objektverz.$dateiname_region);
45 }
46 $datum_differenz = $datum_aktuell[0] - $datum_datei;
47 if ($datum_differenz > 86400 OR !file_exists($objektverz.$dateiname_region)) {
48     /* Daten vom FTP Server holen */
49     $dataload->interhome_ftpload($objektverz, $dateiname_region, $dateinamezip_region);
50     $dateigroesse_region = $dataload->dateigroesse;
51     echo $dataload->info_ftpload;
52 }
53
54 if (file_exists($objektverz.$dateiname_stadt)) {
55     $datum_datei = filemtime($objektverz.$dateiname_stadt);
56 }
57 $datum_differenz = $datum_aktuell[0] - $datum_datei;
58 if ($datum_differenz > 86400 OR !file_exists($objektverz.$dateiname_stadt)) {
59     /* Daten vom FTP Server holen */
60     $dataload->interhome_ftpload($objektverz, $dateiname_stadt, $dateinamezip_stadt);
61     $dateigroesse_stadt = $dataload->dateigroesse;

```

```

62  echo $dataload->info_ftpload;
63 }
64
65 /* Erstellung Array */
66 $dataload->geo-zuweisung($objektverz,$dateiname_land,'land');
67 /* Zweidimensionales Array, in der zweiten Dimension:
68 0 = Landeskürzel, 1 = Regionsnummer, 2 = Regionsname */
69 $land_array = $dataload->landesnamen;
70 $eintraege_land = count($land_array);
71
72 $dataload->geo-zuweisung($objektverz,$dateiname_region,'region');
73 $region_array = $dataload->regionsnamen;
74 $eintraege_region = count($region_array);
75
76 $dataload->geo-zuweisung($objektverz,$dateiname_stadt,'ort');
77 $ort_array = $dataload->ortsnamen;
78 $eintraege_ort = count($ort_array);
79 $tabelle = 'ref_interhome';
80 $i = 0;
81
82 $handle = fopen($objektverz.$dateiname_objekte, "r");
83 $contents = fread($handle, $dateigroesse_objekte);
84 $contentneu = strtr($contents, '"', '_');
85 $zeile = explode("\n", $contentneu);
86 $eintraege=count($zeile);
87 echo $eintraege.'Eingetragene Zeilen in Objects.csv<br>';
88 $ort_auscsv = 0;
89 $ortauscsv_old = 1;
90 while ($i < $eintraege) {
91
92     $spaltenwerte = explode(";", $zeile[$i]);
93     $eintraege_zeile = count($spaltenwerte);
94     $i++;
95     /* $spaltenwerte[1] = Land, $spaltenwerte[2] = Region, $spaltenwerte[3] = Ort */
96     $land_auscsv = $spaltenwerte[1]; //Zweistelliges Landeskürzel
97     $region_auscsv = $spaltenwerte[2]; //Nur Nummerncode
98     $ort_auscsv = $spaltenwerte[3]; //Nur ein Nummerncode
99     $objstadt = addslashes(utf8_decode($spaltenwerte[8])); // Ortsname
100    //Folgendes durch rtrim und ltrim ersetzen
101    $zeichen_stadt = strlen($objstadt);
102    $zeichen_stadt = $zeichen_stadt-2;
103    $objstadt = substr($objstadt,1,$zeichen_stadt);
104    /* ----- Aufspalten Objektcode in Objekt (Haus) und Kategorie (Wohnung) ----- */
105    $objektarray = explode('.', $spaltenwerte[0]);
106    $objektcode = $objektarray[0].'.$objektarray[1];
107    $zimmercode = $objektarray[2];
108    $datensatzvorhanden = FALSE;
109    //Leere Datensätze vermeiden
110    //Wenn ein und derselbe Ort mehrfach nacheinander,
111    //ab 2. Vorkommen nicht mehr die Datenbank quälen
112    if ($ortauscsv_old != $ort_auscsv) {
113        if ($land_auscsv != "" AND $region_auscsv != "" AND $ort_auscsv != "") {
114            /*- Einträge mit Referenztabelle vergleichen. Fehlende Datensätze ergänzen -*/
115            $query_rewrite = "SELECT fh_land, fh_region, fh_place FROM
116            $tabelle WHERE ih_land = '$land_auscsv' AND ih_region = '$region_auscsv' AND
117            ih_place = '$ort_auscsv' ORDER BY fh_region";
118            $aufruf_rewrite = mysql_query($query_rewrite);
119            if (mysql_num_rows($aufruf_rewrite) >= 1) {
120                $datensatzvorhanden = TRUE;
121            }
122            if ($datensatzvorhanden == TRUE) {
123                /* — Variante A, Datensatz konnte mit Referenztabelle abgeglichen werden. —
124                — Eintragen in die CSV Datei — */
125                $datensatzvorhanden = FALSE;
126            }
127            else {

```

```

128      echo '<span style="color:999999;">Keine Übereinstimmung in
129      Referenztabelle für '. $land_auscsv. ', '. $region_auscsv. ', '.
130      $ort_auscsv. 'Suche passenden Landeseintrag in Referenztabelle </span><br>';
131      $query_neudata_land = "SELECT fh_land FROM
132      $tabelle WHERE ih_land = '$land_auscsv' LIMIT 1";
133      $aufruf_neudata_land = mysql_query($query_neudata_land);
134      $zeilen_neudata_land = mysql_num_rows($aufruf_neudata_land);
135      if ($zeilen_neudata_land >= 1) {
136          echo '<span style="color:009900;">Landeseintrag gefunden, Suche nach
137      passender Region</span><br>';
138          $query_neudata_region = "SELECT fh_region FROM $tabelle WHERE
139      ih_land = '$land_auscsv' AND ih_region = '$region_auscsv' LIMIT 1";
140          $aufruf_neudata_region = mysql_query($query_neudata_region);
141          $zeilen_neudata_region = mysql_num_rows($aufruf_neudata_region);
142          if ($zeilen_neudata_region >= 1) {
143              echo '<span style="color:009900;">Passenden Regionseintrag gefunden,
144      Suche passenden Ort, bzw. füge neu ein </span><br>';
145              $query_interhomeref = "SELECT * FROM $tabelle WHERE
146      ih_land = '$land_auscsv' AND ih_region = '$region_auscsv'
147      ORDER BY fh_region";
148              $aufruf_interhomeref = mysql_query($query_interhomeref);
149              while($line_aufruf_interhomeref =
150      mysql_fetch_array($aufruf_interhomeref, MYSQL_ASSOC)) {
151                  $land_ref = $line_aufruf_interhomeref['fh_land'];
152                  $region_ref = $line_aufruf_interhomeref['fh_region'];
153              }
154              $dataload->stammdaten_ortsvergleich($land_ref, $region_ref, $objstadt);
155              echo $dataload->suchmessage;
156              $ort_ref = $dataload->ort_ref;
157
158              /* ——— Eintragen in die Interhome Referenztabelle ——— */
159              $query_neudata_ortinsert = "SELECT * FROM $tabelle WHERE
160      ih_land = '$land_auscsv' AND ih_region = '$region_auscsv'
161      ORDER BY fh_region";
162              $aufruf_neudata_ortinsert = mysql_query($query_neudata_ortinsert);
163              $fhort1_insert = 0;
164              while($line = mysql_fetch_array($aufruf_neudata_ortinsert,
165      MYSQL_ASSOC)) {
166                  $landname_insert = addslashes($line['Land']);
167                  $regionname_insert = addslashes($line['Region']);
168              }
169              $insert_datensatz_ort = "INSERT INTO $tabelle (fh_land, ih_land,
170      Land, fh_region, ih_region, Region, fh_place, ih_place, Ski, Ort)
171      VALUES ('$land_ref', '$land_auscsv', '$landname_insert', '$region_ref',
172      '$region_auscsv', '$regionname_insert', '$ort_ref', '$ort_auscsv',
173      '0', '$objstadt')";
174              if (mysql_query($insert_datensatz_ort)) {
175                  echo '<span style="color:#009900;">Datensatz erfolgreich eingefügt
176      in ref_interhome (Ortssuche, Eintrag)</span><br>';
177              }
178              else {
179                  echo '<span style="color:#990000;">FEHLER: Datensatz konnte nicht
180      in interhome_area eingefügt werden! (Ortssuche, Eintrag)'.
181                  mysql_error(). '</span><br>';
182              }
183          }
184      else { //Wenn Region nicht auffindbar
185          echo '<span style="color:#999900;">Keinen Passenden Regionseintrag
186      gefunden<br>Suche nach passender Region, bzw. füge neu ein ... </span><br>';
187          $query_interhomeref = "SELECT * FROM $tabelle WHERE
188      ih_land = '$land_auscsv' ORDER BY fh_region";
189          $aufruf_interhomeref = mysql_query($query_interhomeref);
190          while ($line_aufruf_interhomeref =
191      mysql_fetch_array($aufruf_interhomeref, MYSQL_ASSOC)) {
192              $land_ref = $line_aufruf_interhomeref['fh_land'];
193          }

```



```

194     $i = 0;
195     $erfolg = 0;
196     while ($erfolg != 1 AND $i < $eintraege-region) {
197         if ($region_array[$i][1] == $region-auscsv AND
198             preg_match("/\b$land-auscsv\b/i", $region_array[$i][0])) {
199             $fh-region-name =
200                 ltrim(rtrim(addslashes(utf8_decode($region_array[$i][2]))));
201             echo '-'. $fh-region-name. '-';
202             $erfolg = 1;
203         }
204         $i++;
205     }
206     /* Regionssuche */
207     $dataloader->stammdaten-regionsvergleich($land-ref, $fh-region-name,
208         $objstadt);
209     echo $dataloader->suchmessage;
210     //Variable in der die letzte aus der Datenbank abgerufene fh
211     //Regionsid aus fh-region steht
212     $region-ref = $dataloader->region-ref;
213     /* Bei Bedarf zus. Ortssuche */
214     if ($dataloader->suchswitch == 'ortsuchen') {
215         $dataloader->stammdaten-ortsvergleich($land-ref, $region-ref, $objstadt);
216         $ort-ref = $dataloader->ort-ref;
217         echo $dataloader->suchmessage;
218     }
219
220     $query-neudata-regioninsert = "SELECT_*_FROM_$tabelle_WHERE
221     ....._ih_land=_'$land-auscsv'_ORDER_BY_fh_region";
222     $aufruf-neudata-regioninsert = mysql_query($query-neudata-regioninsert);
223
224     while($line =
225         mysql_fetch_array($aufruf-neudata-regioninsert, MYSQL_ASSOC)) {
226         $fhland-insert = $line['fh_land'];
227         $landname-insert = $line['Land'];
228     }
229     $insert-datensatz-region = "INSERT INTO_$tabelle_(fh_land, _ih_land,
230     ....._Land, _fh_region, _ih_region, _Region, _fh_place, _ih_place, _Ski, _Ort)
231     .....VALUES(' $fhland-insert ', _'$land-auscsv ', _'$landname-insert ',
232     .....'$region-ref ', _'$region-auscsv ', _'$fh-region-name ', _'$ort-ref ',
233     .....'$ort-auscsv ', _'0', _'$objstadt')";
234     echo $insert-datensatz-region;
235     if (mysql_query($insert-datensatz-region)) {
236         echo '<span style="color:#009900;">Datensatz erfolgreich eingefügt
237     .....(Regionssuche, _Eintrag)</span><br>';
238     }
239     else {
240         echo '<span style="color:#990000;">FEHLER: _Datensatz _konnte _nicht
241     .....eingefügt _werden! _Interhome_area(Regionssuche, _Eintrag)'.
242         mysql_error(). '</span><br>';
243     }
244 }
245 }
246 else { //Else wenn Land nicht auffindbar
247     echo '<span style="color:#999900;">Keinen _Passenden _Landeseintrag
248     .....gefunden, _Suche _nach _passendem _Land _bzw. _füge _komplett _neuen _Datensatz
249     .....ein _(stamm-landessuche)... </span><br>';
250     /* Ermittlung Landesname aus Nummer */
251     $i = 0;
252     $erfolg = 0;
253     while ($erfolg != 1 AND $i < $eintraege-land) {
254         if (preg_match("/\b$land-auscsv\b/i", $land_array[$i][1])) {
255             $fh-land-name =
256                 ltrim(rtrim(addslashes(utf8_decode($land_array[$i][3]))));
257             $erfolg = 1;
258         }
259         $i++;

```

```

260     }
261     $i = 0;
262     $erfolg = 0;
263     while ($erfolg != 1 AND $i < $eintraege-region) {
264         if ($region_array[$i][1] == $region_auscsv AND
265             preg_match("/\b$land_auscsv\b/i", $region_array[$i][0])) {
266             $fh_region_name =
267                 ltrim(rtrim(addslashes(utf8_decode($region_array[$i][2]))));
268             echo 'Regionssuche-aktiv-Regionsname:'. $fh_region_name . '<br>';
269             $erfolg = 1;
270         }
271         $i++;
272     }
273     echo 'fhlandname:'. $fh_land_name . '<br>';
274     echo 'fhregionname:'. $fh_region_name . '<br>';
275
276     $datalog->stammdaten_landesvergleich($fh_land_name, $fh_region_name,
277     $objstadt);
278     echo $datalog->suchmessage;
279     if ($datalog->suchswitch == 'regionsuchen') {
280         $land_ref = $datalog->land_ref;
281         $datalog->stammdaten_regionsvergleich($land_ref, $fh_region_name,
282         $objstadt);
283         $region_ref = $datalog->region_ref;
284         //falls neuer Ort direkt in der Regionssuche eingefügt wird
285         $ort_last = $datalog->ort_ref;
286         echo $datalog->suchmessage;
287
288         if ($datalog->suchswitch == 'ortsuchen') {
289             $datalog->stammdaten_ortsvergleich($land_ref, $region_ref,
290             $objstadt);
291             $ort_ref = $datalog->ort_ref;
292             echo $datalog->suchmessage;
293         }
294     }
295     $ihregion_insert = $region_auscsv;
296     $insert_datensatz_land = "INSERT INTO $tabelle_($fh_land, $ih_land, $Land,
297     $fh_region, $ih_region, $Region, $fh_place, $ih_place, $Ski, $Ort)
298     VALUES(' $land_ref ', ' $land_auscsv ', ' $fh_land_name ', ' $region_ref ',
299     ' $region_auscsv ', ' $fh_region_name ', ' $ort_ref ', ' $ort_auscsv ', '0',
300     ' $objstadt ')";
301     if (mysql_query($insert_datensatz_land)) {
302         echo '<span style="color:#009900;">Datensatz erfolgreich eingefügt
303     (Landessuche, $Eintrag)</span><br>';
304     }
305     else {
306         echo '<span style="color:#990000;">FEHLER: Datensatz konnte nicht
307     eingefügt werden! Interhome-area(Landessuche, $Eintrag)'. mysql_error().
308     '</span><br>';
309     }
310     }
311     }
312     }
313     $ortauscsv_old = $ort_auscsv;
314     }
315 }
316 fclose($handle);
317 mysql_close($db_open);
318 unlink($objektverz.$dateiname_land);
319 unlink($objektverz.$dateiname_region);
320 unlink($objektverz.$dateiname_stadt);
321 unlink($objektverz.$dateiname_objekte);

```

C.11. interhome_stammdaten2_eintragung.php

Listing C.14: Stammdaten - Eintragung in die Datenbank

```

1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/interhome/';
4 $dateinamezip = 'Objects.csv.zip';
5 $dateiname = 'Objects.csv';
6
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $datenbankzugriff;
10 include $dataload_inc;
11
12 $dataload = new dataload;
13 $tabelle_query = 'usr-web31.1.interhome-area';
14 //
15 // Teil 1, Neue CSV Datei erstellen
16 //
17 $anbieter = 1;
18 $strenner = '";"';
19 $strenner2 = '"';
20 $i = 0;
21
22 $dateigroesse = filesize($objektverz.$dateiname);
23
24 /* Neue CSV Datei erstellen */
25 $dataload->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
26 $zeile = $dataload->zeilen;
27 $eintraege = $dataload->eintraege;
28 echo $eintraege.'_Zeilen<br>';
29 $dateiwrite = tempnam($objektverz, 'abc');
30
31 /* ----- Datei Schreiben ----- */
32 if (is_writable($dateiwrite)) {
33     $handle_writefile = fopen($dateiwrite, "a");
34     while ($i < $eintraege) {
35         $spaltenwerte = explode(";", $zeile[$i]);
36         $eintraege_zeile = count($spaltenwerte);
37         $i++;
38     }
39     /* ----- Beschreibung Regions, Orts, Landescodierung ----- */
40     $spaltenwerte[1] = Land, $spaltenwerte[2] = Region, $spaltenwerte[3] = Ort */
41     $land_auscsv = $spaltenwerte[1];
42     $region_auscsv = $spaltenwerte[2];
43     $ort_auscsv = $spaltenwerte[3];
44     /* ----- Aufspalten Objektcode in Objekt (Haus) und Kategorie (Wohnung) ----- */
45     $objektarray = explode('.', $spaltenwerte[0]);
46     $objektcode = $objektarray[0].'.$objektarray[1];
47     $zimmercode = $objektarray[2];
48     if ($land_auscsv != "" AND $region_auscsv != "" AND $ort_auscsv != "") {
49         /* ----- Einträge mit Referenztabelle vergleichen, Identifizierungen
50         nach Ferienhaus Format umschreiben ----- */
51         //Vergleich nach ih_land, ih_region, ih_place, Ort
52         $query_rewrite = "SELECT _fh_land, _fh_region, _fh_place FROM
53         $tabelle_query WHERE _ih_land = '$land_auscsv' AND
54         _ih_region = '$region_auscsv' AND _ih_place = '$ort_auscsv' ORDER BY _fh_region";
55         $aufruf_rewrite = mysql_query($query_rewrite);
56         while ($line_rewrite = mysql_fetch_array($aufruf_rewrite, MYSQL_ASSOC)) {
57             $land_neu = $line_rewrite['fh_land'];
58             $region_neu = $line_rewrite['fh_region'];
59             $ort_neu = $line_rewrite['fh_place'];
60             $eintragen = $strenner2.$anbieter.$strenner.$objektcode.$strenner.$zimmercode.
61             $strenner.$land_neu.$strenner.$region_neu.$strenner.$ort_neu.$strenner.'0'.
62             $strenner2."\\n";
63             fwrite($handle_writefile, $eintragen);
64         }
65     }

```

```
66     }
67     else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
68
69     //-----
70     //Teil2, Einlesen der Daten in die Datenbank
71     //-----
72     $tabelle = 'obj-stammdaten';
73
74     $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
75     echo $dataload->info_indb;
76
77     mysql_close($db_open);
78     fclose($handle_writefile);
79     unlink($dateiwrite);
80
81     $dataload->interhome_backupfile($objektverz, $backupverz, $dateiname);
82     echo $dataload->info_backupfile;
```

D. Aggregationsscripte für Wolters Datensätze

D.1. wolters_bilder.php

Listing D.1: Aggregation von Bildern - Wolters

```

1 $pfad = getenv("DOCUMENT_ROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/wolters/';
4 $dateinamegz = 'bilder.xml.gz';
5 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
6 $dataload_inc = $pfad.'/include/class_dataload.php';
7 include $dataload_inc;
8
9 $trenner = ',';
10 $trenner2 = ' ';
11 $merkmale_max = 0;
12 $anbieter = 2;
13
14 $dataload = new dataload;
15
16 $dateiload = tempnam($objektverz, 'abc');
17
18 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
19 echo $dataload->info_ftpload;
20
21 $dateiwrite = tempnam($objektverz, 'abc');
22
23 if (file_exists($dateiload)) {
24     $xml = simplexml_load_file($dateiload);
25 }
26 else { exit('Warnung, Datei: '.$dateiload.' konnte nicht geladen werden'); }
27
28 if (is_writable($dateiwrite)) {
29     $handle_writefile = fopen($dateiwrite, "a"); //Datei zum anhängen öffnen
30     /* *Sämtliche XML Knoten namens bilder durchgehen* */
31     foreach($xml->bilder as $fbilder) {
32         $i = 1;
33         $objreferenz = $fbilder[objektID];
34         /**Sämtliche XML Knoten namens bild innerhalb des Knotens "bilder" durchgehen**/
35         foreach($fbilder->bild as $einzelbild) {
36             $tuiserverpfad = 'http://www.tui-wolters.de/bilder/Haeuser/300/';
37             $bildlink = $tuiserverpfad.$einzelbild;
38             /* *Erstellung des Eintragsstrings nach dem Schema der Tabelle
39             "pictures_haus" in der Ferienhaus Datenbank* */
40             $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.$trenner.
41             $i.$trenner.'NULL'.$trenner.$bildlink.$trenner2."\\n";
42             /* *Eintragen des neuen Datensatzes anhängend in Datei* */
43             fwrite($handle_writefile, $eintragen);
44             $i++;
45         }
46     }

```

```

47 }
48 else { exit('Warnung, _Datei:_'. $dateiwrite. '_konnte_nicht_gefunden_werden'); }
49 fclose($handle_writefile);
50
51 /* Einlesen in Datenbank */
52 include $datenbankzugriff;
53 $stabelle = 'obj-hauspic';
54 $dataload->stammdaten_indb($stabelle, $dateiwrite, $anbieter);
55 echo $dataload->info_indb;
56 mysql_close($db.open);
57 unlink($dateiload);
58 unlink($dateiwrite);

```

D.2. wolters_buchungshinweise.php

Listing D.2: Aggregation von Buchungshinweisen - Wolters

```

1 $pfad = getenv("DOCUMENTROOT");
2 $jahr = date("y");
3 $objektverz = $pfad. '/objektdaten/';
4 echo $objektverz;
5 $backupverz = $pfad. '/objektdaten/backup/wolters/';
6 $dateinamegz = 'buchungshinweise_de-DE_'. $jahr. '.xml.gz';
7 $dateiname = 'buchungshinweise_de-DE_'. $jahr. '.xml';
8 $datenbankzugriff = $pfad. '/include/opendb_ferienhaus.php';
9 $dataload_inc = $pfad. '/include/class_dataload.php';
10 include $dataload_inc;
11
12 $trenner = '";"';
13 $trenner2 = '""';
14 $merkmale_max = 0;
15 $anbieter = 2;
16
17 $dataload = new dataload;
18
19 $dateiload = tempnam($objektverz, 'abc');
20
21 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
22 echo $dataload->info_ftpload;
23
24 $dateiwrite = tempnam($objektverz, 'abc');
25
26 if (file_exists($dateiload)) {
27     $xml = simplexml_load_file($dateiload);
28 }
29 else { exit('Warnung, _Datei:_'. $dateiload. '_konnte_nicht_geladen_werden'); }
30
31 if (is_writable($dateiwrite)) {
32     $handle_writefile = fopen($dateiwrite, "a"); //Datei zum anhängen öffnen
33     /* *Sämtliche XML Knoten namens bilder durchgehen* */
34     foreach($xml->buchungshinweis as $fbuhinweis) {
35         $i = 1;
36         $objreferenz = (string)$fbuhinweis->objektID;
37         foreach($xml->buchungshinweis->hinweis as $fhinweis) {
38             $datumvon = (string)$fhinweis->datum_von;
39             $datumbis = (string)$fhinweis->datum_bis;
40             $text_D = (string)$fhinweis->text_D;
41             $eintragen = $trenner2. $anbieter. $trenner. $objreferenz. $trenner. ' '. $trenner.
42             $text_D. $trenner. $datumvon. $trenner. $datumbis. $trenner2. "\n";
43             fwrite($handle_writefile, $eintragen);
44             $i++;
45         }
46     }
47 }

```

```

48 else { exit('Warnung, _Datei:_'. $dateiwrite. '_konnte_nicht_gefunden_werden'); }
49 fclose($handle_writefile);
50
51 /* Einlesen in Datenbank */
52 include $datenbankzugriff;
53 $tabelle = 'obj_buhinweise';
54 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
55 echo $dataload->info_indb;
56
57 mysql_close($db_open);
58 unlink($dateiload);
59 unlink($dateiwrite);

```

D.3. wolters_merkmale.php

Listing D.3: Aggregation von Merkmalen - Wolters

```

1 $jahr = date("y");
2 $pfad = getenv("DOCUMENTROOT");
3 $objektverz = $pfad.'/objektdaten/';
4 $backupverz = $pfad.'/objektdaten/backup/wolters/';
5 $dateinamegz = 'merkmale_de_DE_'.$jahr.'.xml.gz';
6 $dateiname = 'merkmale_de_DE_'.$jahr.'.xml';
7 $dateinamegz_pers = 'stammdaten_'.$jahr.'.xml.gz';
8 $dateiname_pers = 'stammdaten_'.$jahr.'.xml';
9 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
10 $dataload_inc = $pfad.'/include/class_dataload.php';
11 include $dataload_inc;
12
13 $trenner = ',';
14 $trenner2 = ' ';
15 $anbieter = 2;
16 $merkmale_max = 0;
17
18 $dataload = new dataload;
19
20 $dateiload_pers = tempnam($objektverz, 'abc');
21 /* Daten vom FTP holen */
22 $dataload->wolters_ftpload($objektverz, $dateiload_pers, $dateinamegz_pers);
23 echo $dataload->info_ftpload;
24 if (file_exists($dateiload_pers)) {
25     $xml_pers = simplexml_load_file($dateiload_pers);
26 }
27 else { exit('Warnung, _Datei:_'. $dateiload_pers. '_konnte_nicht_geladen_werden'); }
28
29 /* Aufbau Zweidimensionales Personen Array 1. Dimension = Datensatz,
30 in der 2. D: 0 = objref, 1 = objkat, 2 = anz. Personen, 3 = anz. Kinder */
31 $i = 0;
32 foreach($xml_pers->objekt as $fobjekt) {
33
34     $objreferenz = ltrim(rtrim((string)$fobjekt->objektID));
35     $objkategorie = ltrim(rtrim((string)$fobjekt->categoryID));
36     $personen = ltrim(rtrim((string)$fobjekt->max_person));
37     $kinder = ltrim(rtrim((string)$fobjekt->nr_of_childs));
38     $personen_array[$i] = array($objreferenz, $objkategorie, $personen, $kinder);
39     $i++;
40 }
41 unlink($dateiload_pers);
42 $personen_zeilen = count($personen_array);
43 /* ENDE Arrayerstellung */
44
45 $dateiload = tempnam($objektverz, 'abc');
46 /* Daten vom FTP holen */
47 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);

```

```

48 echo $dataload->info.ftpload;
49
50 $dateiwrite = tempnam($objektverz, 'abc');
51 /* Daten in temporäre csv datei übernehmen */
52 if (file_exists($dateiload)) {
53     $xml = simplexml_load_file($dateiload);
54 }
55 else { exit('Warnung, \Datei: \'. $dateiload. \' konnte nicht geladen werden'); }
56
57 if (is_writable($dateiwrite)) {
58     $handle_writefile = fopen($dateiwrite, "w");
59     fclose($handle_writefile);
60     $handle_writefile = fopen($dateiwrite, "a"); //Datei zum anhängen öffnen
61     /* *Sämtliche XML Knoten namens merkmale durchgehen* */
62     foreach($xml->merkmale as $fmerkmale) {
63         $merkmalanzahl = count($fmerkmale->merkmal);
64         $objreferenz = $fmerkmale['objektID'];
65         $objektschluesel = $fmerkmale['categoryID'];
66
67         $i = 0; $erfolg = 0; $personen_anz = 0; $kinder_anz = 0;
68         while($erfolg != 1 AND $i < $personen_zeilen) {
69             if (preg_match("/\b$objreferenz\b/i", $personen_array[$i][0]) AND
70                 preg_match("/\b$objektschluesel\b/i", $personen_array[$i][1])) {
71                 $personen_anz = $personen_array[$i][2];
72                 $kinder_anz = $personen_array[$i][3];
73             }
74             $i++;
75         }
76
77         /*fluschen der vorherigen Arrayeinträge, 35 = Anzahl aller möglichen Merkmale*/
78         for ($i = 0; $i < 35; $i++) {
79             $merkmale_sammeln[$i] = 0;
80         }
81         /* *Datengewinnung aus den Merkmalsknoten* */
82         for ($i = 0; $i < $merkmalanzahl; $i++) {
83             $merkmal = utf8_decode((string)$fmerkmale->merkmal[$i]);
84             $merkmal_wert = (string)$fmerkmale->merkmal[$i]['wert'];
85             /* *Notwendige Fallunterscheidung da Position der Merkmale variiert* */
86             if ($merkmal == 'Anlage') {
87                 $merkmale_sammeln[0] = 1;
88             }
89
90             if ($merkmal == 'Anz. \Haustiere') {
91                 $merkmale_sammeln[1] = $merkmal_wert;
92             }
93
94             if ($merkmal == 'Anzahl \Schlafräume') {
95                 $merkmale_sammeln[2] = $merkmal_wert;
96             }
97
98             if ($merkmal == 'Badelandschaft') {
99                 $merkmale_sammeln[3] = 1;
100             }
101
102             if ($merkmal == 'Boot') {
103                 $merkmale_sammeln[4] = 1;
104             }
105
106             if ($merkmal == 'Caravanpark') {
107                 $merkmale_sammeln[5] = 1;
108             }
109
110             if ($merkmal == 'Endreinigung_inkl.') {
111                 $merkmale_sammeln[6] = 1;
112             }
113

```



```
114     if ($merkmal == 'Entf._Angelmgf.') {
115         $merkmale_sammeln[7] = $merkmal_wert;
116     }
117
118     if ($merkmal == 'Entf._Bademgl.') {
119         $merkmale_sammeln[8] = $merkmal_wert;
120     }
121
122     if ($merkmal == 'Entf._Fjord_') {
123         $merkmale_sammeln[9] = $merkmal_wert;
124     }
125
126     if ($merkmal == 'Entf._Strand') {
127         $merkmale_sammeln[10] = $merkmal_wert;
128     }
129
130     if ($merkmal == 'Entf._Ski_Loipe') {
131         $merkmale_sammeln[11] = $merkmal_wert;
132     }
133
134     if ($merkmal == 'Flexible_Anreise') {
135         $merkmale_sammeln[12] = 1;
136     }
137
138     if ($merkmal == 'gem._Außenpool') {
139         $merkmale_sammeln[13] = 1;
140     }
141
142     if ($merkmal == 'gem._Innenpool') {
143         $merkmale_sammeln[14] = 1;
144     }
145
146     if ($merkmal == 'Garage') {
147         $merkmale_sammeln[15] = 1;
148     }
149
150     if ($merkmal == 'Geschirrspüler') {
151         $merkmale_sammeln[16] = 1;
152     }
153
154     if ($merkmal == 'Golf') {
155         $merkmale_sammeln[17] = 1;
156     }
157
158     if ($merkmal == 'Haus') {
159         $merkmale_sammeln[18] = 1;
160     }
161
162     if ($merkmal == 'Kamin-/Ofen') {
163         $merkmale_sammeln[19] = 1;
164     }
165
166     if ($merkmal == 'Kinderbetreuung') {
167         $merkmale_sammeln[20] = 1;
168     }
169
170     if ($merkmal == 'Miniferien') {
171         $merkmale_sammeln[21] = 1;
172     }
173
174     if ($merkmal == 'Nachbarhäuser_vorhanden') {
175         $merkmale_sammeln[22] = 1;
176     }
177
178     if ($merkmal == 'Objektgröße_in_Qm') {
179         $merkmale_sammeln[23] = $merkmal_wert;
```

```

180     }
181
182     if ($merkmal == 'Privatpool') {
183         $merkmale_sammeln[24] = 1;
184     }
185
186     if ($merkmal == 'Rollstuhlfahrerfreundlich') {
187         $merkmale_sammeln[25] = 1;
188     }
189
190     if ($merkmal == 'SAT-TV') {
191         $merkmale_sammeln[26] = 1;
192     }
193
194     if ($merkmal == 'Sauna') {
195         $merkmale_sammeln[27] = 1;
196     }
197
198     if ($merkmal == 'Serviceeinrichtungen') {
199         $merkmale_sammeln[28] = 1;
200     }
201
202     if ($merkmal == 'Solarium') {
203         $merkmale_sammeln[29] = 1;
204     }
205
206     if ($merkmal == 'Waschmaschine') {
207         $merkmale_sammeln[30] = 1;
208     }
209
210     if ($merkmal == 'Wellness & Beauty') {
211         $merkmale_sammeln[31] = 1;
212     }
213
214     if ($merkmal == 'Whirl-Pool') {
215         $merkmale_sammeln[32] = 1;
216     }
217
218     if ($merkmal == 'Wohnung') {
219         $merkmale_sammeln[33] = 1;
220     }
221 }
222
223 if ($merkmalanzahl > $merkmale_max) {
224     $merkmale_max = $merkmalanzahl;
225 }
226
227 /* *Erstellung des Eintragsstrings nach dem Schema der Tabelle "merkmale"
228 in der Ferienhaus Datenbank vom September 2008* */
229 $eintragen = $trenner2.$anbieter.$trenner.$objreferenz.$trenner.
230 $objektschlüssel.$trenner.'0'.$trenner.$merkmale_sammeln[0].
231 $trenner.$merkmale_sammeln[3].$trenner.$merkmale_sammeln[25].$trenner.
232 $merkmale_sammeln[4].$trenner.$merkmale_sammeln[5].$trenner.$merkmale_sammeln[6].
233 $trenner.$merkmale_sammeln[15].$trenner.$merkmale_sammeln[16].$trenner.
234 '0'.$trenner.$merkmale_sammeln[17].$trenner.$merkmale_sammeln[1].$trenner.
235 $merkmale_sammeln[19].$trenner.$merkmale_sammeln[20].$trenner.
236 $merkmale_sammeln[14].$trenner.$merkmale_sammeln[27].$trenner.
237 $merkmale_sammeln[3].$trenner.'0'.$trenner.'0'.$trenner.$personen_anz.$trenner.
238 $skinder_anz.$trenner.'0'.$trenner.$merkmale_sammeln[2].$trenner.
239 $merkmale_sammeln[29].$trenner.$merkmale_sammeln[26].$trenner.'0'.
240 $trenner.'0'.$trenner.$merkmale_sammeln[30].$trenner.'0'.$trenner.'0'.$trenner.
241 $merkmale_sammeln[31].$trenner.$merkmale_sammeln[32].$trenner.
242 $merkmale_sammeln[7].$trenner.$merkmale_sammeln[8].$trenner.
243 $merkmale_sammeln[9].$trenner.$merkmale_sammeln[11].$trenner.
244 $merkmale_sammeln[10].$trenner.$merkmale_sammeln[12].$trenner.
245 $merkmale_sammeln[21].$trenner.'0'.$trenner.'0'.$trenner.'0'.$trenner.'0'.
246 $trenner.'0'.$trenner.'0'.$trenner.'0'.$trenner.'0'.$trenner.'0'.

```

```

246     $strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.
247     '0'.$strenner.'0'.$strenner2."\n";
248     /* *Eintragen des neuen Datensatzes anhängend in Datei* */
249     fwrite($handle_writefile, $eintragen);
250 }
251 }
252 else { exit('Warnung, \Datei: \'.$dateiwrite.\ \'konnte nicht gefunden werden'); }
253 fclose($handle_writefile);
254
255 /* Daten in Datenbank einlesen */
256 include $datenbankzugriff;
257 $tabelle = 'obj-merkmale';
258
259 $datalog->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
260 echo $datalog->info_indb;
261
262 mysql_close($db_open);
263 unlink($dateiload);
264 unlink($dateiwrite);

```

D.4. wolters_preise.php

Listing D.4: Aggregation von Preisen - Wolters

```

1 $jahr = date("y");
2 $pfad = getenv("DOCUMENT_ROOT");
3 $objektverz = $pfad.'/objektdaten/';
4 $backupverz = $pfad.'/objektdaten/backup/wolters/';
5 $dateinamegz = 'kontingent_'.$jahr.'.xml.gz';
6 $dateiname = 'kontingent_'.$jahr.'.xml';
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $datalog_inc = $pfad.'/include/class_datalog.php';
9 include $datalog_inc;
10
11 $strenner = '";';
12 $strenner2 = '";';
13 $anbieter = 2;
14
15 //Hinweist: Intern für Wolters abfrageroutine Status DB: I = inaktiv (
16 //Datensatz nicht übernehmen), A,D = aktiv
17 $datalog = new datalog;
18
19 $dateiload = tempnam($objektverz, 'abc');
20
21 $datalog->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
22 echo $datalog->info_ftpload;
23
24 $dateiwrite = tempnam($objektverz, 'abc');
25
26 if (file_exists($dateiload)) {
27     $xml = simplexml_load_file($dateiload);
28 }
29 else { exit('Warnung, \Datei: \'.$dateiload.\ \'konnte nicht geladen werden'); }
30
31 if (is_writable($dateiwrite)) {
32     $handle_writefile = fopen($dateiwrite, "a"); //Datei zum anhängen öffnen
33
34     /* *Sämtliche XML Knoten namens bilder durchgehen* */
35     foreach($xml->Kontingent as $fkontingent) {
36         $status_objekt = $fkontingent['StatusDB'];
37         if ($status_objekt != 'I') {
38             $objreferenz = $fkontingent->objektID;
39             $kategorie = $fkontingent->categoryID;
40             $datumvon = $fkontingent->datumKZ;

```

```

41     $preis7tage = $fkontingent->preis_1W;
42     $preis14tage = $fkontingent->preis_2W;
43     $preis21tage = $fkontingent->preis_3W;
44     $preis28tage = $fkontingent->preis_4W;
45
46     /* *Erstellung des Eintragsstrings nach dem Schema der Tabelle
47     "fh-preise" in der Ferienhaus Datenbank* */
48     $eintragen1 = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.
49     $kategorie.$strenner.'7'.$strenner.$datumvon.$strenner.'NULL'.
50     $strenner.$preis7tage.$strenner2."\n";
51     $eintragen2 = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.
52     $kategorie.$strenner.'14'.$strenner.$datumvon.$strenner.'NULL'.
53     $strenner.$preis14tage.$strenner2."\n";
54     $eintragen3 = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.
55     $kategorie.$strenner.'21'.$strenner.$datumvon.$strenner.'NULL'.
56     $strenner.$preis21tage.$strenner2."\n";
57     $eintragen4 = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.
58     $kategorie.$strenner.'28'.$strenner.$datumvon.$strenner.'NULL'.
59     $strenner.$preis28tage.$strenner2."\n";
60
61     /* *Eintragen des neuen Datensatzes anhängend in Datei* */
62     fwrite($handle_writefile, $eintragen1);
63     fwrite($handle_writefile, $eintragen2);
64     fwrite($handle_writefile, $eintragen3);
65     fwrite($handle_writefile, $eintragen4);
66 }
67 }
68 }
69 else { exit('Warnung, _Datei:_'.$dateiwrite.'_konnte_nicht_gefunden_werden'); }
70 fclose($handle_writefile);
71
72 /* Einlesen in die Datenbank */
73 include $datenbankzugriff;
74 $tabelle = 'obj-preise';
75
76 $datalog->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
77 echo $datalog->info_indb;
78
79 mysql_close($db_open);
80 unlink($dateiload);
81 unlink($dateiwrite);

```

D.5. wolters_hausprosa.php

Listing D.5: Aggregation der Hausbeschreibungen - Wolters

```

1  $jahr = date("y");
2  $pfad = getenv("DOCUMENTROOT");
3  $objektverz = $pfad.'/objektdaten/';
4  $backupverz = $pfad.'/objektdaten/backup/wolters/';
5  $dateinamegz = 'prosa_de_DE_'.$jahr.'.xml.gz';
6  $dateiname = 'prosa_de_DE_'.$jahr.'.xml';
7  $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8  $datalog_inc = $pfad.'/include/class_datalog.php';
9  include $datalog_inc;
10
11 $strenner = '";"';
12 $strenner2 = '""';
13 $anbieter = 2;
14 $i=0;
15 $slashed = "/";
16
17 $datalog = new datalog;
18 $dateiload = tempnam($objektverz, 'abc');

```

```

19 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
20 echo $dataload->info_ftpload;
21 $dateiwrite = tempnam($objektverz, 'abc');
22
23 if (file_exists($dateiload)) {
24 $xml = simplexml_load_file($dateiload);
25 }
26 else { exit('Konnte_' . $dateiload . '_nicht_öffnen. '); }
27
28 if (is_writable($dateiwrite)) {
29 $handle_writefile = fopen($dateiwrite, "w");
30 fclose($handle_writefile);
31 //Datei zum anhängen öffnen
32 $handle_writefile = fopen($dateiwrite, "a");
33
34 //Jedes XML Objekt namens "objekt" abarbeiten
35 foreach($xml->objekt as $fobjekt) {
36 $catidvor = $fobjekt->objektID;
37 if (!preg_match($slashed.$catidvor, $catidvor2)) {
38 /* geht nacheinander alle Text Childs des
39 jeweiligen Objekt Knotens durch */
40 foreach ($fobjekt->text AS $beschrtext) {
41 if ($beschrtext['topic'] == 'beschreibung' AND
42 $beschrtext['nr'] == '1') {
43 $desc_haus1 = $beschrtext;
44 }
45 if ($beschrtext['topic'] == 'entfernungen' AND
46 $beschrtext['nr'] == '1') {
47 $desc_entfernung1 = $beschrtext;
48 }
49 }
50 $eintragen = $strenner2.$anbieter.$strenner.$catidvor.$strenner.'de'.
51 $strenner.$desc_haus1.'<br><br>'.$desc_entfernung1.$strenner2.'\n';
52 fwrite($handle_writefile, $eintragen);
53 }
54 $catidvor2 = $fobjekt->objektID;
55 }
56 }
57 else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
58 fclose($handle_writefile);
59
60 /* In die Datenbank eintragen */
61 include $datenbankzugriff;
62 $tabelle = 'obj_haustext';
63 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
64 echo $dataload->info_indb;
65
66 mysql_close($db_open);
67 unlink($dateiload);
68 unlink($dateiwrite);

```

D.6. wolters_zimmerprosa.php

Listing D.6: Aggregation der Wohnungsbeschreibungen - Wolters

```

1 $jahr = date("y");
2 $pfad = getenv("DOCUMENT_ROOT");
3 $objektverz = $pfad.'/objektdaten/';
4 $backupverz = $pfad.'/objektdaten/backup/wolters/';
5 $dateinamegz = 'prosa_de_DE_'.$jahr.'.xml.gz';
6 $dateiname = 'prosa_de_DE_'.$jahr.'.xml';
7 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $dataload_inc;

```

```

10
11 $strenner = '";';
12 $strenner2 = '";';
13 $anbieter = 2;
14 $i=0;
15 $slashed = "/";
16
17 $dataload = new dataload;
18 $dateiload = tempnam($objektverz, 'abc');
19 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
20 echo $dataload->info_ftpload;
21 $dateiwrite = tempnam($objektverz, 'abc');
22
23 if (file_exists($dateiload)) {
24     $xml = simplexml_load_file($dateiload);
25 }
26 else { exit('Konnte_test.xml_nicht_öffnen. '); }
27
28 if (is_writable($dateiwrite)) {
29     $handle_writefile = fopen($dateiwrite, "w");
30     fclose($handle_writefile);
31     //Datei zum anhängen öffnen
32     $handle_writefile = fopen($dateiwrite, "a");
33
34     //Jedes XML Objekt namens "merkmale" abarbeiten
35     foreach($xml->objekt as $fobjekt) {
36         /**
37          * $eigenschaften
38          * [0] = $anbieter;
39          * //Angabe "string" Wandelt Objekt in String um
40          * [1] = (string)$merkmale['objektID'];
41          * //Entfernung vom Strand in 'int' gewandelt,
42          * //Wert liegt innerhalb Array1 im Array wert
43          * [14] = (string)$merkmale->merkmal[1]['wert']+0;
44          */
45         // $i = count($merkmale->merkmal);
46         $ferienobjekt = $fobjekt->objektID;
47         $objektschlüssel = $fobjekt->categoryID;
48         /* geht nacheinander alle Text Childs des
49          jeweiligen Objekt Knotens durch */
50         foreach ($fobjekt->text AS $beschrtext) {
51             if ($beschrtext['topic'] == 'kategoriebeschreibung' AND
52                 $beschrtext['nr'] == '1') {
53                 $desc_wohnung1 = $beschrtext;
54             }
55             if ($beschrtext['topic'] == 'zubs' AND $beschrtext['nr'] == '1') {
56                 $desc_zubs1 = $beschrtext;
57             }
58             if ($beschrtext['topic'] == 'rabatte' AND $beschrtext['nr'] == '1') {
59                 $desc_rabatte1 = $beschrtext;
60             }
61         }
62         $eintragen = $strenner2.$anbieter.$strenner.$ferienobjekt.$strenner.
63         $objektschlüssel.$strenner.'de'.$strenner.'Wohnung:.'. $desc_wohnung1.
64         '_Zusatzangebot:.'. $desc_zubs1. '_Rabatte:.'. $desc_rabatte1.$strenner2."\n";
65         fwrite($handle_writefile, $eintragen);
66     }
67 }
68 else { echo 'Datei_konnte_zum_Schreiben_nicht_geoeffnet_werden'; }
69 fclose($handle_writefile);
70
71 /* In die Datenbank eintragen */
72 include $datenbankzugriff;
73 $tabelle = 'obj-wohnungstext';
74 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
75 echo $dataload->info_indb;

```

```

76
77 mysql_close($db_open);
78 unlink($dateiload);
79 unlink($dateiwrite);

```

D.7. wolters_zubs.php

Listing D.7: Aggregation von Zusatzleistungen - Wolters

```

1 $jahr = date("y");
2 $pfad = getenv("DOCUMENT_ROOT");
3 $objektverz = $pfad.'/objektdaten/';
4 $backupverz = $pfad.'/objektdaten/backup/wolters/';
5 $dateinamegz = 'zubs_de_DE_'.$jahr.'.xml.gz';
6 $dateiname = 'zubs_de_DE_'.$jahr.'.xml';
7 $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $dataload_inc;
10
11 $strenner = '";';
12 $strenner2 = '";';
13 $merkmale_max = 0;
14 $anbieter = 2;
15
16 $dataload = new dataload;
17
18 $dateiload = tempnam($objektverz, 'abc');
19
20 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
21 echo $dataload->info_ftpload;
22
23 $dateiwrite = tempnam($objektverz, 'abc');
24
25 if (file_exists($dateiload)) {
26     $xml = simplexml_load_file($dateiload);
27 }
28 else { exit('Warnung, „Datei: „'.$dateiload.'“ konnte nicht geladen werden'); }
29
30 if (is_writable($dateiwrite)) {
31     $handle_writefile = fopen($dateiwrite, "w");
32     fclose($handle_writefile);
33     //Datei zum anhängen öffnen
34     $handle_writefile = fopen($dateiwrite, "a");
35
36     /* *Sämtliche XML Knoten namens bilder durchgehen* */
37     foreach($xml->zubs as $fzubs) {
38         $i = 1;
39         $objreferenz = $fzubs->objektID;
40         $kategorie = $fzubs->categoryID;
41
42         /* *Sämtliche XML Knoten namens bild innerhalb
43         des Knotens "bilder" durchgehen* */
44         foreach($fzubs->zub as $leistung) {
45             $leistungsname = $leistung->zubName;
46             $leistungseinheit = $leistung->unit;
47             $leistungspreis = $leistung->price;
48             /* *Erstellung des Eintragsstrings nach dem Schema der Tabelle
49             "pictures_haus" in der Ferienhaus Datenbank* */
50             $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.
51             $kategorie.$strenner.$leistungsname.$strenner.$leistungspreis.
52             $strenner.'EUR'.$strenner.' '.$strenner.$leistungseinheit.$strenner2."\n";
53             /* *Eintragen des neuen Datensatzes anhängend in Datei* */
54             fwrite($handle_writefile, $eintragen);
55             $i++;

```

```

56     }
57 }
58 }
59 else { exit('Warnung, _Datei:_'.$dateiwrite.'
60 _konnte_nicht_gefunden_werden'); }
61 fclose($handle_writefile);
62
63 /* Einlesen in Datenbank */
64 include $datenbankzugriff;
65 $tabelle = 'obj_zusatzleistungen';
66
67 $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
68 echo $dataload->info_indb;
69
70 mysql_close($db_open);
71 unlink($dateiload);
72 unlink($dateiwrite);

```

D.8. wolters_stammdaten1_referenzierung.php

Listing D.8: Referenzierung der Stammdaten - Wolters

```

1 $jahr = date("y");
2 $pfad = getenv("DOCUMENT_ROOT");
3 $objektverz = $pfad.'/objektdaten/';
4 $backupverz = $pfad.'/objektdaten/backup/wolters/';
5 $dateinamegz = 'stammdaten_'.$jahr.'.xml.gz';
6 $dateiname = 'stammdaten_'.$jahr.'.xml';
7 $dateinamegeogz = 'area_de_DE.xml.gz';
8 $dateinamegeo = 'area_de_DE.xml';
9 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
10 $dataload_inc = $pfad.'/include/class_dataload.php';
11 include $dataload_inc;
12 $tabelle = 'ref_wolters';
13
14 $dataload = new dataload;
15
16 $dateiload = tempnam($objektverz, 'abc');
17 $dateiload_geodaten = tempnam($objektverz, 'abc');
18
19 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
20 echo $dataload->info_ftpload;
21 $dataload->wolters_ftpload($objektverz, $dateiload_geodaten,
22 $dateinamegeogz);
23 echo $dataload->info_ftpload;
24
25 $trenner = '";'";
26 $trenner2 = '";';
27 $anbieter = 2;
28 $i=0;
29 $slashed = "/";
30 include $datenbankzugriff;
31
32 /* Länder und Regionen mit ihren Kürzeln im Array erfassen */
33 $fh_region_name = 0;
34 if (file_exists($dateiload_geodaten)) {
35     $xml_geo = simplexml_load_file($dateiload_geodaten);
36 }
37 else { exit('Konnte_'.$dateiload_geodaten.'_nicht_öffnen. '); }
38
39 $i = 0;
40 $geo_region_old = '_';
41 //Jedes XML Objekt namens "area" abarbeiten
42 foreach($xml_geo->area as $farea) {

```



```

43 $regionsname =
44 ltrim(rtrim(utf8_decode(addslashes((string)$farea->region_D))));
45 if ($geo_region_old != $regionsname) {
46     $xml_geo_region[$i] = array((string)$farea['region_kz'], $regionsname);
47     $geo_region_old = $regionsname;
48     $i++;
49 }
50 }
51 $regionszeilen = count($xml_geo_region);
52
53 $i = 0;
54 $geo_land_old = '_';
55 //Jedes XML Objekt namens "area" abarbeiten
56 foreach($xml_geo->area as $farea) {
57     $landesname =
58     ltrim(rtrim(utf8_decode(addslashes((string)$farea->land_D))));
59     if ($geo_land_old != $landesname) {
60         $xml_geo_land[$i] = array((string)$farea['land_kz'], $landesname);
61         $geo_land_old = $landesname;
62         $i++;
63     }
64 }
65 $landeszeilen = count($xml_geo_land);
66
67 if (file_exists($dateiload)) {
68     $xml = simplexml_load_file($dateiload);
69 }
70 }
71 else { exit('Konnte_' . $dateiload . '_nicht_öffnen. '); }
72
73 /* *Sämtliche XML Knoten namens objekt durchgehen* */
74 //Jedes XML Objekt namens "objekt" abarbeiten
75 foreach($xml->objekt as $fobjekt) {
76
77     /* Deklaration Objektdaten */
78     $land = $fobjekt['land_kz'];
79     $region = $fobjekt['region_kz'];
80     $area = $fobjekt['area_kz']; //Landes, Regions, Gebietsname
81
82     $objreferenz = $fobjekt->objektID;
83     $objstadt = utf8_decode($fobjekt->city);
84     $objstadt = addslashes($objstadt);
85     $objstadt = rtrim(ltrim($objstadt));
86
87     //Wenn ein und derselbe Ort mehrfach nacheinander
88     //ab 2. Vorkommen nicht mehr die Datenbank quälen
89     if ($objstadt_old != $objstadt) {
90         /* Leere Datensätze ausschließen */
91         if ($land != "" AND $region != "" AND $objstadt != "") {
92
93             /* *Einträge mit Referenztabelle vergleichen,
94             Fehlende Einträge ergänzen* */
95             //Vergleich nach wo_land, wo_region, wo_places, Ort
96             $query_rewrite = "SELECT _fh_land, _fh_region, _fh_city FROM
97             .....$tabelle WHERE _wo_land = '$land' AND _wo_region = '$region' AND
98             .....wo_places = $area AND _Ort = '$objstadt' ORDER BY _fh_region";
99             $aufruf_rewrite = mysql_query($query_rewrite);
100             while ($line_rewrite =
101             mysql_fetch_array($aufruf_rewrite, MYSQL_ASSOC)) {
102                 $datensatzvorhanden = TRUE;
103             }
104
105             if ($datensatzvorhanden == TRUE) {
106                 /* *Variante A, Datensatz konnte mit Referenztabelle
107                 abgeglichen werden. — Eintragen in die CSV Datei* */
108                 $datensatzvorhanden = FALSE;

```

```

109     }
110     else {
111         echo '<span style="color:999999;">Keine Übereinstimmung in
112     Referenztabelle für ',$land.',',$region.',',$area.
113     '$_Suche$_passenden Landeseintrag in Referenztabelle </span><br>';
114     $query_neudata_land = "SELECT fh_land FROM $tabelle WHERE
115     wo_land=$_'$land' LIMIT 1";
116     $aufruf_neudata_land = mysql_query($query_neudata_land);
117     $zeilen_neudata_land = mysql_num_rows($aufruf_neudata_land);
118     if ($zeilen_neudata_land >= 1) {
119         echo '<span style="color:009900;">Landeseintrag gefunden, $_Suche
120     nach passender Region</span><br>';
121         $query_neudata_region = "SELECT fh_region FROM $tabelle WHERE
122     wo_land=$_'$land' AND wo_region=$_'$region' LIMIT 1";
123         $aufruf_neudata_region = mysql_query($query_neudata_region);
124         $zeilen_neudata_region = mysql_num_rows($aufruf_neudata_region);
125         if ($zeilen_neudata_region >= 1) {
126             echo '<span style="color:009900;">Passenden Regionseintrag
127     gefunden, $_Suche$_passenden Ort, bzw. füge neu ein</span><br>';
128             /* ferienhaus.de Landes ID und Regions ID holen
129             aus wolters_area */
130             $query_woltersref = "SELECT fh_land, fh_region FROM
131     $tabelle WHERE wo_land=$_'$land' AND wo_region=$_'$region'
132     ORDER BY fh_region";
133             $aufruf_woltersref = mysql_query($query_woltersref);
134             while ($line_aufruf_woltersref =
135     mysql_fetch_array($aufruf_woltersref, MYSQL_ASSOC)) {
136                 $land_ref = $line_aufruf_woltersref['fh_land'];
137                 $region_ref = $line_aufruf_woltersref['fh_region'];
138             }
139
140             $dataload->wostammdaten.ortsvergleich($land_ref,
141             $region_ref, $objstadt);
142             echo $dataload->suchmessage;
143             $ort_last = $dataload->ort_ref;
144
145             /* *Eintragen in die Wolters Referenztabelle* */
146             $query_neudata_ortinsert = "SELECT * FROM $tabelle WHERE
147     wo_land=$_'$land' AND wo_region=$_'$region' ORDER BY fh_region";
148             $aufruf_neudata_ortinsert = mysql_query($query_neudata_ortinsert);
149             $fhregion_insert = 0;
150             $fhplaces_insert = 0;
151             $fhcity_insert = 0;
152             $fhortl_insert = 0;
153             while($line =
154     mysql_fetch_array($aufruf_neudata_ortinsert, MYSQL_ASSOC)) {
155                 $landname_insert = $line['Land'];
156                 $regionname_insert = $line['Region'];
157                 $woregion_insert = $line['wo_region'];
158                 $placesname_insert = $line['Place'];
159                 if ($line['fh_places'] >= $fhplaces_insert) {
160                     $fhplaces_insert = $line['fh_places'];
161                 }
162             }
163             $insert_datensatz_ort = "INSERT INTO $tabelle (fh_land,
164     wo_land, Land, fh_region, wo_region, Region, fh_places,
165     wo_places, Place, fh_city, Ort)
166     VALUES ('$land_ref', '$land', '$landname_insert', '$region_ref',
167     '$woregion_insert', '$regionname_insert', '$fhplaces_insert',
168     '$area', '$placesname_insert', '$ort_last', '$objstadt')";
169             if (mysql_query($insert_datensatz_ort)) {
170                 echo '<span style="color:#009900;">Datensatz erfolgreich
171     eingefügt in wolters_area (Ortssuche, Eintrag)</span><br>';
172             }
173             else {
174                 echo '<span style="color:#990000;">FEHLER: Datensatz konnte

```

```

175 .....nicht_in_wolters_area_eingefügt_werden!_(Ortssuche, _Eintrag)'.
176     mysql_error() . '</span><br>';
177 }
178 }
179 else { //Else wenn Region nicht auffindbar
180     echo '<span style="color:#999900;">Keinen_Passenden
181 .....Regionseintrag_gefunden<br>Suche_nach_passender_Region ,
182 .....bzw_Füge_neu_ein...</span><br>';
183     $query_woltersref = "SELECT fh_land FROM $tabelle WHERE
184 .....wo_land=_ '$land' ORDER BY fh_region";
185     $aufruf_woltersref = mysql_query($query_woltersref);
186     while ($line_aufruf_woltersref =
187 .....mysql_fetch_array($aufruf_woltersref, MYSQL_ASSOC)) {
188         $land_ref = $line_aufruf_woltersref['fh_land'];
189     }
190     $i = 0;
191     $erfolg = 0;
192     while ($i < $regionszeilen AND $erfolg != 1) {
193         if (preg_match("/\b$region\b/i", $xml_geo_region[$i][0])) {
194             $fh_region_name = $xml_geo_region[$i][1];
195             $erfolg = 1;
196         }
197         $i++;
198     }
199     $dataload->wostammdaten_regionsvergleich($land_ref,
200     $fh_region_name, $objstadt);
201     echo $dataload->suchmessage;
202     $region_ref = $dataload->region_ref;
203     if ($dataload->suchswitch == 'ortsuchen') {
204         $dataload->wostammdaten_ortsvergleich($land_ref,
205         $region_ref, $objstadt);
206         $ort_last = $dataload->ort_ref;
207         echo $dataload->suchmessage;
208     }
209
210     $query_neudata_regioninsert = "SELECT * FROM $tabelle WHERE
211 .....wo_land=_ '$land' ORDER BY fh_region";
212     $aufruf_neudata_regioninsert =
213     mysql_query($query_neudata_regioninsert);
214     $fhregion_insert = 0; $fhplaces_insert = 0; $fhcity_insert = 0;
215     while($line =
216 .....mysql_fetch_array($aufruf_neudata_regioninsert, MYSQL_ASSOC)) {
217         $landname_insert = $line['Land'];
218         if ($line['fh_places'] >= $fhplaces_insert) {
219             $fhplaces_insert = $line['fh_places'];
220         }
221     }
222     $fhplaces_insert = $fhplaces_insert+1;
223     $woregion_insert = $region;
224     echo '$ort_last_in_RegionNEU:_' . $ort_last . '<br>';
225     $insert_datensatz_region = "INSERT INTO $tabelle (fh_land ,
226 .....wo_land, _Land, _fh_region, _wo_region, _Region, _fh_places ,
227 .....wo_places, _Place, _fh_city, _Ort)
228 .....VALUES(' $land_ref ', _ '$land' , _ '$landname_insert ', _ '$region_ref ',
229 ..... '$woregion_insert ', _ '$fh_region_name ', _ '$fhplaces_insert ',
230 ..... '$area ', _ 'NULL ', _ '$ort_last ', _ '$objstadt ')" ;
231     if (mysql_query($insert_datensatz_region)) {
232         echo '<span style="color:#009900;">Datensatz_erfolgreich
233 .....eingefügt (Regionssuche, _Eintrag)</span><br>';
234     }
235     else {
236         echo '<span style="color:990000;">FEHLER: _Datensatz
237 .....konnte_nicht_eingefügt_werden! _Wolters_area (Regionssuche ,
238 .....Eintrag)'. mysql_error() . '</span><br>';
239     }
240 }

```

```

241     }
242     else { //Else wenn Land nicht auffindbar
243         echo '<span style="color:#999900;">Keinen Passenden Landeseintrag
244         .....gefunden, Suche nach passendem Land bzw. füge komplett neuen
245         .....Datensatz ein (stamm.landessuche)...</span><br>';
246         $i = 0;
247         $erfolg = 0;
248         while($i < $landeszeilen AND $erfolg != 1) {
249             if (preg_match("/\b$land\b/i", $xml_geo_land[$i][0])) {
250                 $fh_land_name = $xml_geo_land[$i][1];
251                 $erfolg = 1;
252             }
253             $i++;
254         }
255         $i = 0;
256         $erfolg = 0;
257         while($i < $regionszeilen AND $erfolg != 1) {
258             if (preg_match("/\b$region\b/i", $xml_geo_region[$i][0])) {
259                 $fh_region_name = $xml_geo_region[$i][1];
260                 $erfolg = 1;
261             }
262             $i++;
263         }
264         echo 'fhlandname:'. $fh_land_name. '<br>';
265         echo 'fhregionname:'. $fh_region_name. '<br>';
266         $dataload->wostammdaten_landesvergleich($fh_land_name,
267         $fh_region_name, $objstadt);
268         echo $dataload->suchmessage;
269         if ($dataload->suchswitch == 'regionsuchen') {
270             $land_ref = $dataload->land_ref;
271             $dataload->wostammdaten_regionsvergleich($land_ref,
272             $fh_region_name, $objstadt);
273             $region_ref = $dataload->region_ref;
274             //falls neuer Ort direkt in der Regionssuche eingefügt wird
275             $ort_last = $dataload->ort_ref;
276             echo $dataload->suchmessage;
277             if ($dataload->suchswitch == 'ortsuchen') {
278                 $dataload->wostammdaten_ortsvergleich($land_ref,
279                 $region_ref, $objstadt);
280                 $ort_last = $dataload->ort_ref;
281                 echo $dataload->suchmessage;
282             }
283         }
284
285         $query_neudata_landinsert = "SELECT_*_FROM_$tabelle
286         .....ORDER_BY_fh_region";
287         $aufruf_neudata_landinsert =
288         mysql_query($query_neudata_landinsert);
289         while($line = mysql_fetch_array($aufruf_neudata_landinsert,
290         MYSQL_ASSOC)) {
291             $fhregion_insert = $line['fh_region'];
292         }
293         $woregion_insert = $region;
294         $fhplaces_insert = '0';
295
296         $insert_datensatz_land = "INSERT INTO_$tabelle_(fh_land, _wo_land,
297         .....Land, _fh_region, _wo_region, _Region, _fh_places, _wo_places, _Place,
298         .....fh_city, _Ort)
299         .....VALUES('$_land_ref', '$_$land', '$_$fh_land_name', '$_$region_ref',
300         .....$_$woregion_insert, '$_$fh_region_name', '$_$fhplaces_insert',
301         .....$_$area, '$_NULL', '$_$ort_last', '$_$objstadt)";
302         if (mysql_query($insert_datensatz_land)) {
303             echo '<span style="color:#009900;">Datensatz erfolgreich
304             .....eingefügt (Landessuche, Eintrag)</span><br>';
305         }
306         else {

```

```

307         echo '<span style="color:#990000;">FEHLER: Datensatz konnte
308 nicht eingefügt werden! Wolters_area(Landessuche, Eintrag)'.
309         mysql_error(). '</span><br>';
310     }
311 }
312 }
313 }
314 $fobjektende = $fobjekt;
315 $objstadt_old = $objstadt;
316 }
317 }
318 mysql_close($db_open);

```

D.9. wolters_stammdaten2_eintragung.php

Listing D.9: Aggregation von Stammdaten - Wolters

```

1 $jahr = date("y");
2 $pfad = getenv("DOCUMENTROOT");
3 $objektverz = $pfad.'/objektdaten/';
4 $backupverz = $pfad.'/objektdaten/backup/wolters/';
5 $dateinamegz = 'stammdaten_'.$jahr.'.xml.gz';
6 $dateiname = 'stammdaten_'.$jahr.'.xml';
7 $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
8 $dataload_inc = $pfad.'/include/class_dataload.php';
9 include $dataload_inc;
10 include $datenbankzugriff;
11 $stabelle_query = 'ref-wolters';
12
13 $trenner = ',';
14 $trenner2 = ' ';
15 $anbieter = 2;
16 $dataload = new dataload;
17
18 $dateiload = tempnam($objektverz, 'abc');
19
20 $dataload->wolters_ftpload($objektverz, $dateiload, $dateinamegz);
21 echo $dataload->info_ftpload;
22
23 $dateiwrite = tempnam($objektverz, 'abc');
24 //-----
25 //Teil1, CSV Datei erstellen
26 //-----
27
28 if (file_exists($dateiload)) {
29     $xml = simplexml_load_file($dateiload);
30 }
31 else { exit('Warnung: Datei: '.$dateiload.' konnte nicht geladen werden'); }
32 if (is_writable($dateiwrite)) {
33     $handle_writefile = fopen($dateiwrite, "w");
34     fclose($handle_writefile);
35     //Datei zum anhängen öffnen
36     $handle_writefile = fopen($dateiwrite, "a");
37     /* *Sämtliche XML Knoten namens objekt durchgehen* */
38     //Jedes XML Objekt namens "merkmale" abarbeiten
39     foreach($xml->objekt as $fobjekt) {
40         /* Deklaration Objektdaten */
41         $land = $fobjekt['land_kz'];
42         $region = $fobjekt['region_kz'];
43         $area = $fobjekt['area_kz'];
44         $objreferenz = $fobjekt->objektID;
45         $objektschluesel = $fobjekt->categoryID;
46         $objstadt1 = utf8_decode($fobjekt->city);
47         $objstadt = addslashes($objstadt1);

```

```

48  /* Leere Datensätze ausschließen */
49  /* als inaktiv markierte Datensätze auch nicht übernehmen */
50  if ($land != "" AND $region != "" AND $area != "") {
51      $query_eintragen = "SELECT fh_land, fh_region, fh_city FROM
52  ----- $tabelle_query WHERE wo_land = '$land' AND wo_region = '$region' AND
53  ----- wo_places = '$area' AND Ort = '$objstadt' ORDER BY fh_region";
54      $aufruf_eintragen = mysql_query($query_eintragen);
55      /* *Ersetzen von Land, Regions Stadtcodes* */
56      while ($line_eintragen =
57  mysql_fetch_array($aufruf_eintragen, MYSQL_ASSOC)) {
58          $eintragen = $strenner2.$anbieter.$strenner.$objreferenz.$strenner.
59          $objektschlüssel.$strenner.$line_eintragen['fh_land'].$strenner.
60          $line_eintragen['fh_region'].$strenner.$line_eintragen['fh_city'].$
61          $strenner2."\n";
62          /* *Eintragen des neuen Datensatzes anhängend in Datei* */
63          fwrite($handle_writefile, $eintragen);
64      }
65  }
66  }
67  }
68  else { echo 'Datei konnte zum Schreiben nicht geöffnet werden'; }
69  fclose($handle_writefile);
70
71  //-----
72  //Teil2, Einlesen der Daten in die Datenbank
73  //-----
74
75  /* In die Datenbank eintragen */
76  $tabelle = 'obj_stammdaten';
77  $dataload->stammdaten_indb($tabelle, $dateiwrite, $anbieter);
78  echo $dataload->info_indb;
79
80  mysql_close($db_open);
81  unlink($dateiload);
82  unlink($dateiwrite);

```

E. Aggregationsscripte ursprüngliches auf neues System

E.1. Allgemeine Vorgehensweise

Eine Datenaufbereitung ist nicht nötig. Da alle Daten bereits während des Exports aufbereitet werden erfolgt als letzter Schritt lediglich das Einlesen in die Datenbank.

Listing E.1: Aggregation allgemeine Vorgehensweise

```
1 $pfad = getenv("DOCUMENTROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/onferienhaus/';
4 $dateiname = 'texte.csv';
5
6 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
7 $dataload_inc = $pfad.'/include/class_dataload.php';
8 include $dataload_inc;
9
10 $anbieter = 99;
11
12 $dataload = new dataload;
13
14 /* Daten vom FTP Server holen */
15 $dataload->onferienhaus_ftpload($objektverz, $dateiname);
16 $dateigroesse = $dataload->dateigroesse;
17 echo 'Dateigroesse'. $dateigroesse;
18 echo $dataload->info_ftpload;
19
20 /* einlesen Daten in die Datenbank */
21 include $datenbankzugriff;
22 $tabelle = 'obj_haustext';
23
24 $dataload->stammdaten_indb($tabelle, $objektverz.$dateiname, $anbieter);
25 echo $dataload->info_indb;
26
27 mysql_close($db-open);
```

E.2. onferienhaus_stammdaten1_referenzierung

Listing E.2: Referenzierung Stammdaten altes System

```
1 $pfad = getenv("DOCUMENTROOT");
2
3 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
4
5 $objektverz = $pfad.'/objektdaten/';
6 $dateiname_objekte = 'stammdaten.csv';
7
```

```

8 include $datenbankzugriff;
9 $dataload_inc = $pfad.'/include/class_dataload.php';
10 include $dataload_inc;
11
12 $dataload = new dataload;
13 $datum_aktuell = getdate();
14
15 $dataload->onferienhaus_ftpload($objektverz, $dateiname_objekte);
16 $dateigroesse_objekte = $dataload->dateigroesse;
17 echo $dataload->info_ftpload;
18
19 $tabelle = 'ref_onferienhaus';
20 $i = 0;
21
22 $handle = fopen($objektverz.$dateiname_objekte, "r");
23 $contents = fread($handle, $dateigroesse_objekte);
24 $contentneu = strtr($contents, '"', '_');
25 $zeile = explode("\n", $contentneu);
26 $eintraege=count($zeile);
27 echo $eintraege.'Eingetragene Zeilen in Stammdaten.csv<br>';
28 $ort_auscsv = 0;
29 $ortauscsv_old = 1;
30 while ($i < $eintraege) {
31
32     $spaltenwerte = explode(";", $zeile[$i]);
33     $eintraege_zeile = count($spaltenwerte);
34     $i++;
35     /* $spaltenwerte[1] = Land, $spaltenwerte[2] = Region, $spaltenwerte[3] = Ort */
36     //Landesname
37     $land_auscsv = addslashes(utf8_decode(ltrim(rtrim($spaltenwerte[3]))));
38     //Regionsname
39     $region_auscsv = addslashes(utf8_decode(ltrim(rtrim($spaltenwerte[4]))));
40     //Ortsname
41     $ort_auscsv = addslashes(utf8_decode(ltrim(rtrim($spaltenwerte[5]))));
42     // Ortsname
43     $objstadt = addslashes(utf8_decode(ltrim(rtrim($spaltenwerte[5]))));
44     echo 'Land'. $land_auscsv. '_Region'. $region_auscsv. '_Ort'. $ort_auscsv. '<br>';
45     $objektcode = $spaltenwerte[1];
46     $zimmercode = '0';
47     $datensatzvorhanden = FALSE;
48     //Leere Datensätze vermeiden
49     if ($ortauscsv_old != $ort_auscsv) { //Wenn ein und derselbe Ort mehrfach
50     //nacheinander ab 2. Vorkommen nicht mehr die Datenbank quälen
51         if ($land_auscsv != "" AND $region_auscsv != "" AND $ort_auscsv != "") {
52             /* - Einträge mit Referenztabelle vergleichen. Fehlende Datensätze ergänzen - */
53             $query_rewrite = "SELECT fh_land, fh_region, fh_place FROM $tabelle WHERE
54             Land=$land_auscsv AND Region=$region_auscsv AND Ort=$ort_auscsv
55             ORDER BY fh_region";
56             $aufruf_rewrite = mysql_query($query_rewrite);
57             if (mysql_num_rows($aufruf_rewrite) >= 1) {
58                 $datensatzvorhanden = TRUE;
59             }
60             if ($datensatzvorhanden == TRUE) {
61                 /* -Variante A, Datensatz konnte mit Referenztabelle abgeglichen werden.
62                 — Eintragen in die CSV Datei— */
63                 $datensatzvorhanden = FALSE;
64             }
65             else {
66                 echo '<span style="color:999999;">Keine Übereinstimmung in Referenztabelle
67                 für '. $land_auscsv. ', '. $region_auscsv. ', '. $ort_auscsv. '_Suche passenden
68                 Landeseintrag in Referenztabelle</span><br>';
69                 $query_neudata_land = "SELECT fh_land FROM $tabelle WHERE
70                 Land=$land_auscsv LIMIT 1";
71                 $aufruf_neudata_land = mysql_query($query_neudata_land);
72                 $zeilen_neudata_land = mysql_num_rows($aufruf_neudata_land);
73                 if ($zeilen_neudata_land >= 1) {

```



```

74         echo '<span style="color:009900;">Landeseintrag gefunden ,
75         Suche nach passender Region</span><br>';
76         $query_neudata_region = "SELECT fh_region FROM $tabelle WHERE
77         Land = '$land_auscsv' AND Region = '$region_auscsv' LIMIT 1";
78         $aufruf_neudata_region = mysql_query($query_neudata_region);
79         $zeilen_neudata_region = mysql_num_rows($aufruf_neudata_region);
80         if ($zeilen_neudata_region >= 1) {
81             echo '<span style="color:009900;">Passenden Regionseintrag gefunden ,
82             Suche passenden Ort, bzw. füge neu ein</span><br>';
83             $query_interhomeref = "SELECT * FROM $tabelle WHERE
84             Land = '$land_auscsv' AND Region = '$region_auscsv' ORDER BY fh_region";
85             $aufruf_interhomeref = mysql_query($query_interhomeref);
86             while($line_aufruf_interhomeref = mysql_fetch_array(
87             $aufruf_interhomeref, MYSQL_ASSOC)) {
88                 $land_ref = $line_aufruf_interhomeref['fh_land'];
89                 $region_ref = $line_aufruf_interhomeref['fh_region'];
90             }
91             $dataload->onstammdaten_ortsvergleich($land_ref, $region_ref, $objstadt);
92             echo $dataload->suchmessage;
93             $ort_ref = $dataload->ort_ref;
94
95             /* ----- Eintragen in die Interhome Referenztabelle ----- */
96             $query_neudata_ortinsert = "SELECT * FROM $tabelle WHERE
97             Land = '$land_auscsv' AND Region = '$region_auscsv' ORDER BY fh_region";
98             $aufruf_neudata_ortinsert = mysql_query($query_neudata_ortinsert);
99             $hort1_insert = 0;
100             while($line = mysql_fetch_array($aufruf_neudata_ortinsert, MYSQL_ASSOC)) {
101                 $landname_insert = addslashes($line['Land']);
102                 $regionname_insert = addslashes($line['Region']);
103             }
104             $insert_datensatz_ort = "INSERT INTO $tabelle (fh_land, on_land,
105             Land, fh_region, on_region, Region, fh_place, on_place, Place,
106             fh_city, Ort)
107             VALUES ('$land_ref', '0', '$landname_insert', '$region_ref', '0',
108             '$regionname_insert', '$ort_ref', '0', '0', '0', '$objstadt')";
109             if (mysql_query($insert_datensatz_ort)) {
110                 echo '<span style="color:#009900;">Datensatz erfolgreich eingefügt
111             in ferienhaus_area (Ortssuche, Eintrag)</span><br>';
112             }
113             else {
114                 echo '<span style="color:#990000;">FEHLER: Datensatz konnte nicht
115             in ferienhaus_area eingefügt werden! (Ortssuche, Eintrag)'.
116                 mysql_error(). '</span><br>';
117             }
118         }
119         else { //Wenn Region nicht auffindbar
120             echo '<span style="color:#999900;">Keinen Passenden Regionseintrag
121             gefunden<br>Suche nach passender Region, bzw. füge neu ein...</span><br>';
122             $query_interhomeref = "SELECT * FROM $tabelle WHERE
123             Land = '$land_auscsv' ORDER BY fh_region";
124             $aufruf_interhomeref = mysql_query($query_interhomeref);
125             while ($line_aufruf_interhomeref = mysql_fetch_array(
126             $aufruf_interhomeref, MYSQL_ASSOC)) {
127                 $land_ref = $line_aufruf_interhomeref['fh_land'];
128             }
129             $fh_region_name = $region_auscsv;
130
131             /* $i = 0;
132             $erfolg = 0;
133             while ($erfolg != 1 AND $i < $eintraege_region) {
134                 if ($region_array[$i][1] == $region_auscsv AND
135                 preg_match("/\b$land_auscsv\b/i", $region_array[$i][0])) {
136                     $fh_region_name = ltrim(rtrim(addslashes(utf8_decode(
137                     $region_array[$i][2]))));
138                     echo '- '$fh_region_name.' -';
139                     $erfolg = 1;

```

```

140     }
141     $i++;
142 } */
143 /* Regionssuche */
144 $datalog->onstammdaten_regionsvergleich($land_ref, $fh_region_name,
145     $objstadt);
146 echo $datalog->suchmessage;
147 $region_ref = $datalog->region_ref; //Variable in der die letzte
148 //aus der Datenbank abgerufene fh Regionid aus fh_region steht
149 /* Bei Bedarf zus. Ortssuche */
150 if ($datalog->suchswitch == 'ortsuchen') {
151     $datalog->onstammdaten_ortsvergleich($land_ref, $region_ref,
152     $objstadt);
153     $ort_ref = $datalog->ort_ref;
154     echo $datalog->suchmessage;
155 }
156 $query_neudata_regioninsert = "SELECT_*_FROM_$tabelle_WHERE
157 Land=$_'$land_auscsv'_ORDER_BY_fh_region";
158 $aufruf_neudata_regioninsert = mysql_query($query_neudata_regioninsert);
159
160 while($line = mysql_fetch_array($aufruf_neudata_regioninsert,
161 MYSQL_ASSOC)) {
162     $fhland_insert = $line['fh_land'];
163     $landname_insert = $line['Land'];
164 }
165 $insert_datensatz_region = "INSERT INTO_$tabelle_(fh_land, _on_land,
166 Land, _fh_region, _on_region, _Region, _fh_place, _on_place, _Place,
167 fh_city, _Ort)
168 VALUES('$_fhland_insert', $_'0', $_'$landname_insert', $_'$region_ref',
169 '0', $_'$fh_region_name', $_'$ort_ref', $_'0', $_'0', $_'0', $_'$objstadt')";
170 echo $insert_datensatz_region;
171 if (mysql_query($insert_datensatz_region)) {
172     echo '<span style="color:#009900;">Datensatz erfolgreich eingefügt
173 (Regionssuche, _Eintrag)</span><br>';
174 }
175 else {
176     echo '<span style="color:#990000;">FEHLER: _Datensatz konnte nicht
177 eingefügt werden! _Interhome.area (Regionssuche, _Eintrag)'.
178 mysql_error(). '</span><br>';
179 //echo '<font color="#FF0000">'. $insert_datensatz_region. '</font><br>';
180 }
181 }
182 }
183 else { //Else wenn Land nicht auffindbar
184     echo '<span style="color:#999900;">Keinen _Passenden _Landeseintrag
185 gefunden, _Suche nach _passendem _Land bzw. _füge _komplett _neuen _Datensatz _ein
186 (stamm_landessuche)...</span><br>';
187 /* Ermittlung Landesname aus Nummer */
188 $fh_land_name = $land_auscsv;
189 $fh_region_name = $region_auscsv;
190 echo 'fhlandname: _'. $fh_land_name. '<br>';
191 echo 'fhregionname: _'. $fh_region_name. '<br>';
192
193 $datalog->onstammdaten_landesvergleich($fh_land_name,
194 $fh_region_name, $objstadt);
195 echo $datalog->suchmessage;
196 if ($datalog->suchswitch == 'regionsuchen') {
197
198     $land_ref = $datalog->land_ref;
199     $datalog->onstammdaten_regionsvergleich($land_ref,
200     $fh_region_name, $objstadt);
201     $region_ref = $datalog->region_ref;
202     $ort_last = $datalog->ort_ref; //falls neuer Ort direkt in der
203     //Regionssuche eingefügt wird
204     echo $datalog->suchmessage;
205

```

```

206         if ($dataload->suchswitch == 'ortsuchen') {
207             $dataload->onstammdaten_ortsvergleich($land_ref,
208             $region_ref, $objstadt);
209             $ort_ref = $dataload->ort_ref;
210             echo $dataload->suchmessage;
211         }
212     }
213     $ihregion_insert = $region_auscsv;
214     $insert_datensatz_land = "INSERT INTO $tabelle_ (fh_land, _on_land,
215     _Land, _fh_region, _on_region, _Region, _fh_place, _on_place, _Place,
216     _fh_city, _Ort)
217     VALUES ('$land_ref', _'0', _'$fh_land_name', _'$region_ref', _'0',
218     _'$fh_region_name', _'$ort_ref', _'0', _'0', _'0', _'$objstadt')";
219     if (mysql_query($insert_datensatz_land)) {
220         echo ' <span style="color:#009900;">Datensatz erfolgreich eingefügt
221     (Landessuche, _Eintrag)</span><br>';
222     }
223     else {
224         echo ' <span style="color:#990000;">FEHLER: _Datensatz konnte nicht
225     eingefügt werden! _Interhome_area(Landessuche, _Eintrag)'.
226     mysql_error(). ' </span><br>';
227     }
228     }
229     }
230     }
231     $ortauscsv_old = $ort_auscsv;
232     }
233 }
234 fclose ($handle);
235 mysql_close ($db_open);

```

E.3. onferienhaus_stammdaten2_eintragung

Listing E.3: Eintragung Stammdaten des alten Systems

```

1 $pfad = getenv("DOCUMENT_ROOT");
2 $objektverz = $pfad.'/objektdaten/';
3 $backupverz = $pfad.'/objektdaten/backup/onferienhaus/';
4 $dateiname = 'stammdaten.csv';
5
6 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
7 $dataload_inc = $pfad.'/include/class_dataload.php';
8 include $datenbankzugriff;
9 include $dataload_inc;
10
11 $dataload = new dataload;
12 $tabelle_query = 'ref_onferienhaus';
13 //-----
14 //Teil1, Neue CSV Datei erstellen
15 //-----
16 $anbieter = 99;
17 $trenner = ',';
18 $trenner2 = ' ';
19 $i = 0;
20
21 $dateigroesse = filesize($objektverz.$dateiname);
22
23 /* Neue CSV Datei erstellen */
24 $dataload->interhome_explodefile($objektverz, $dateiname, $dateigroesse);
25 $zeile = $dataload->zeilen;
26 $eintraege = $dataload->eintraege;
27 echo $eintraege.' _Zeilen<br>';
28 $dateiwrite = tempnam($objektverz, 'abc');
29

```

```

30 /* ----- Datei Schreiben ----- */
31 if (is_writable($dateiwrite)) {
32     $handle_writefile = fopen($dateiwrite, "a");
33     while ($i < $eintraege) {
34         $spaltenwerte = explode(";", $zeile[$i]);
35         $eintraege_zeile = count($spaltenwerte);
36         $i++;
37 /* ----- Beschreibung Regions, Orts, Landescodierung ----- */
38 /* $spaltenwerte[1] = Land, $spaltenwerte[2] = Region, $spaltenwerte[3] = Ort */
39     $objref = ltrim(rtrim($spaltenwerte[1]));
40     $objkat = ltrim(rtrim($spaltenwerte[2]));
41     $land_auscsv = ltrim(rtrim(addslashes(utf8_decode($spaltenwerte[3]))));
42     $region_auscsv = ltrim(rtrim(addslashes(utf8_decode($spaltenwerte[4]))));
43     $ort_auscsv = ltrim(rtrim(addslashes(utf8_decode($spaltenwerte[5]))));
44 /* ----- Aufspalten Objektcode in Objekt (Haus) und Kategorie (Wohnung) ----- */
45     if ($land_auscsv != "" AND $region_auscsv != "" AND $ort_auscsv != "") {
46 /* -Einträge mit Referenztabelle vergleichen, Identifizierungen nach Ferienhaus
47 Format umschreiben- */
48         //Vergleich nach ih_land, ih_region, ih_place, Ort
49         $query_rewrite = "SELECT fh_land, fh_region, fh_place FROM
50 $stabelle_query WHERE Land=' $land_auscsv ' AND
51 $region_auscsv ' $region_auscsv ' AND Ort=' $ort_auscsv ' ORDER BY fh_region";
52         $aufruf_rewrite = mysql_query($query_rewrite);
53         while ($line_rewrite = mysql_fetch_array($aufruf_rewrite, MYSQL_ASSOC)) {
54             $land_neu = $line_rewrite['fh_land'];
55             $region_neu = $line_rewrite['fh_region'];
56             $ort_neu = $line_rewrite['fh_place'];
57             $eintragen = $strenner2.$anbieter.$strenner.$objref.$strenner.$objkat.$strenner.$land_neu.$strenner.$region_neu.$strenner.$ort_neu.$strenner2."\n";
58             fwrite($handle_writefile, $eintragen);
59             echo $eintragen."<br>";
60         }
61     }
62 }
63 }
64 }
65 else { echo 'Datei konnte zum Schreiben nicht geöffnet werden'; }
66
67 //-----
68 //Teil2, Einlesen der Daten in die Datenbank
69 //-----
70 $stabelle = 'obj_stammdaten';
71
72 $dataload->stammdaten_indb($stabelle, $dateiwrite, $anbieter);
73 echo $dataload->info_indb;
74
75 mysql_close($db_open);
76 fclose($handle_writefile);
77 unlink($dateiwrite);
78
79 $dataload->interhome_backupfile($objektverz, $backupverz, $dateiname);
80 echo $dataload->info_backupfile;

```

E.4. hoteltocsv.php

Listing E.4: Exportskript für Daten des alten Systems

```

1 /* init */
2 $strenner = ',';
3 $strenner2 = ' ';
4 $idanbieter = 99;
5 // für Vakanzen
6 $stabelle_vak = 'hotel_belegungsuebersicht';
7 $jahr = date("Y");
8 $monat = date("m");

```

```

9 $tag = date("d");
10 $datum = date("Y-m-d");
11
12
13 $downward = "";
14 $Pfad = getenv("REQUEST_URI");
15 $Pfadteile = explode("/", $Pfad);
16 for($i=count($Pfadteile)-1;$i>=1;$i--)
17 {
18     $downward = $downward . " ../";
19 }
20
21 $includepfad = $downward."ferienhausAdmininclude/" ;
22
23 include ( $includepfad."db_connect.php" ) ;
24
25 $i = 0;
26
27 $dateiwrite_texte = "texte.csv";
28 $dateiwrite_merkmale = "merkmale.csv";
29 $dateiwrite_bilder = "bilder.csv";
30 $dateiwrite_stammdaten = "stammdaten.csv";
31 $dateiwrite_vakanzen = "vakanzen.csv";
32
33 /* Status der Datei überprüfen und bei erfolgreichem Zugriff leeren */
34 if (is_writable($dateiwrite_texte)) {
35     $handle_writefile_texte = fopen($dateiwrite_texte, "w");
36     fclose($handle_writefile_texte);
37 }
38 else {
39     echo'Datei_'. $dateiwrite_texte. '_konnte_zum_Schreiben_nicht_geoeffnet_werden<br>
40     _breche_weitere_Verarbeitung_ab!! ' ;
41     mysql_close_db($db_open);
42     exit();
43 }
44 if (is_writable($dateiwrite_merkmale)) {
45     $handle_writefile_merkmale = fopen($dateiwrite_merkmale, "w");
46     fclose($handle_writefile_merkmale);
47 }
48 else {
49     echo'Datei_'. $dateiwrite_merkmale. '_konnte_zum_Schreiben_nicht_geoeffnet_werden<br>
50     _breche_weitere_Verarbeitung_ab!! ' ;
51     mysql_close_db($db_open);
52     exit();
53 }
54 if (is_writable($dateiwrite_bilder)) {
55     $handle_writefile_bilder = fopen($dateiwrite_bilder, "w");
56     fclose($handle_writefile_bilder);
57 }
58 else {
59     echo'Datei_'. $dateiwrite_bilder. '_konnte_zum_Schreiben_nicht_geoeffnet_werden<br>
60     _breche_weitere_Verarbeitung_ab!! ' ;
61     mysql_close_db($db_open);
62     exit();
63 }
64 if (is_writable($dateiwrite_stammdaten)) {
65     $handle_writefile_stammdaten = fopen($dateiwrite_stammdaten, "w");
66     fclose($handle_writefile_stammdaten);
67 }
68 else {
69     echo'Datei_'. $dateiwrite_stammdaten. '_konnte_zum_Schreiben_nicht_geoeffnet_werden
70     <br>breche_weitere_Verarbeitung_ab!! ' ;
71     mysql_close_db($db_open);
72     exit();
73 }
74 if (is_writable($dateiwrite_vakanzen)) {

```

```

75  $handle_writefile_vakanzen = fopen($dateiwrite_vakanzen , "w");
76  fclose($handle_writefile_vakanzen);
77 }
78 else {
79  echo 'Datei '. $dateiwrite_vakanzen . '_konnte_zum_Schreiben_nicht_geoeffnet_werden<br>
80  _breche_weitere_Verarbeitung_ab!! ' ;
81  mysql_close_db($db.open);
82  exit();
83 }
84 //chmod ($dateiwrite_texte , 0777);
85
86 $query = "SELECT_*_FROM_hotels_ORDER_BY_id_hotel";
87 $abruf = mysql_query($query);
88
89 /* Daten abrufen und Dateien neu befüllen */
90 $handle_writefile_texte = fopen($dateiwrite_texte , "a");
91 $handle_writefile_merkmale = fopen($dateiwrite_merkmale , "a");
92 $handle_writefile_bilder = fopen($dateiwrite_bilder , "a");
93 $handle_writefile_stammdaten = fopen($dateiwrite_stammdaten , "a");
94 $handle_writefile_vakanzen = fopen($dateiwrite_vakanzen , "a");
95
96 while($abruf_line = mysql_fetch_array($abruf, MYSQL_ASSOC)) {
97  $id_land = $abruf_line['id_land'];
98  $id_region = $abruf_line['id_region'];
99  $sortsname = ltrim(rtrim(addslashes(utf8_encode($abruf_line['ort']))));
100  $id_hotel = $abruf_line['id_hotel'];
101  $objekttyp = ltrim(rtrim($abruf_line['objekttyp']));
102  $beschreibung = utf8_encode(addslashes($abruf_line['beschreibung']));
103  $besonderheiten = $abruf_line['besonderheiten'];
104  /* Wandlung in ein neues Format
105  id_hotel
106  1 = ONFH000001, 2 = ONFH000002 usw.
107  objekttyp
108  Haus = HA, Wohnung = WH, Ferienhaus = FH usw. */
109
110  /* id_hotel -> id_hotel_new */
111  $id_hotel_new = '';
112  $nummernlaenge = strlen($id_hotel);
113  $nullserie = 6-$nummernlaenge;
114  for ($k = 0; $k < $nullserie; $k++) {
115    $id_hotel_new = $id_hotel_new.'0';
116  }
117  $id_hotel_new = 'ONFH'. $id_hotel_new. $id_hotel;
118
119
120  /* —Privathauesser Hotelbeschreibung— */
121
122  /* Objekttyp -> Objektkategorie */
123  switch ($objekttyp) {
124    case 'Haus':
125      $objekttyp = 'HA';
126      break;
127    case 'Wohnung':
128      $objekttyp = 'WH';
129      break;
130    case 'Feriendorf':
131      $objekttyp = 'FD';
132      break;
133    case '':
134      $objekttyp = 'KA';
135      break;
136  }
137  $text = $beschreibung."\\n". $besonderheiten;
138
139  $hausbeschreibung_eintragen = $trenner2.$idanbieter.$trenner.
140  $id_hotel_new.$trenner.'de'.$trenner.$text.$trenner2."\\n";

```

```
141 fwrite($handle_writefile_texte , $hausbeschreibung_eintragen);
142
143
144 /* — Privathäuser Merkmale — */
145 $betten = $abruf_line['betten'];
146 $personen = $abruf_line['kapazitaeten_erwachsene'];
147 $kinder = $abruf_line['kapazitaeten_kinder'];
148 $schlafzimmer = $abruf_line['anzahlzimmer_schlafzimmer'];
149 if ($abruf_line['haustiere'] == 'ja') {
150     $haustiere = 1;
151 }
152 else { $haustiere = 0; }
153
154 if ($abruf_line['wellness'] == 'ja') {
155     $wellness = 1;
156 }
157 else { $wellness = 0; }
158
159 if ($abruf_line['pool'] == 'ja') {
160     $pool = 1;
161 }
162 else { $pool = 0; }
163
164 if ($abruf_line['behindertengerecht'] == 'ja') {
165     $rollstuhl = 1;
166 }
167 else { $rollstuhl = 0; }
168 //echo 'Rollstuhl: '. $rollstuhl;
169
170 if ($abruf_line['kinderbetreuung'] == 'ja') {
171     $kinderbetreuung = 1;
172 }
173 else { $kinderbetreuung = 0; }
174
175 if ($abruf_line['ausstattung_kueche_geschirrspuerer'] == 'ja') {
176     $geschirrspuerer = 1;
177 }
178 else { $geschirrspuerer = 0; }
179
180 if ($abruf_line['ausstattung_badezimmer_whirlpool'] == 'ja') {
181     $whirlpool = 1;
182 }
183 else { $whirlpool = 0; }
184
185 if ($abruf_line['ausstattung_kueche_geschirrspuerer'] == 'ja') {
186     $geschirrspuerer = 1;
187 }
188 else { $geschirrspuerer = 0; }
189
190 if ($abruf_line['schwimmbad_innen'] == 'ja' OR
191     $abruf_line['schwimmbad_aussen'] == 'ja') {
192     $schwimmbad = 1;
193 }
194 else { $schwimmbad = 0; }
195
196 if ($abruf_line['offener_kamin'] == 'ja' OR
197     $abruf_line['ausstattung_allg_kamin'] == 'ja') {
198     $kamin = 1;
199 }
200 else { $kamin = 0; }
201
202 if ($abruf_line['waschmaschine'] == 'ja' OR
203     $abruf_line['ausstattung_allg_waschmaschine'] == 'ja') {
204     $waschmaschine = 1;
205 }
206 else { $waschmaschine = 0; }
```

```
207
208   if ($abruf_line['ausstattung-wohnraum_tv'] == 'ja' OR
209   $abruf_line['ausstattung-wohnraum_sattv'] == 'ja') {
210       $tv = 1;
211   }
212   else { $tv = 0; }
213
214   if ($abruf_line['ausstattung-aussenbereich-garage'] == 'ja' OR
215   $abruf_line['ausstattung-aussenbereich-tiefgarage'] == 'ja') {
216       $garage = 1;
217   }
218   else { $garage = 0; }
219
220   if ($abruf_line['klimaanlage'] == 'ja') {
221       $klimaanlage = 1;
222   }
223   else { $klimaanlage = 0; }
224
225   if ($abruf_line['www'] == 'ja' OR $abruf_line['www-wireless'] == 'ja') {
226       $internet = 1;
227   }
228   else { $internet = 0; }
229
230   if ($abruf_line['ausstattung-kueche-kuehlschrank'] == 'ja' OR
231   $abruf_line['ausstattung-kueche-gefrierschrank'] == 'ja') {
232       $kuehlschrank = 1;
233   }
234   else { $kuehlschrank = 0; }
235
236   if ($abruf_line['ausstattung-kueche-herd'] == 'ja' OR
237   $abruf_line['ausstattung-kueche-backofen'] == 'ja') {
238       $herd = 1;
239   }
240   else { $herd = 0; }
241
242   if ($abruf_line['ausstattung-wohnraum-gesellschaftsspiele'] == 'ja') {
243       $gesellschaftsspiele = 1;
244   }
245   else { $gesellschaftsspiele = 0; }
246
247   if ($abruf_line['nichtraucher-zimmer'] == 'ja') {
248       $nichtraucher = 1;
249   }
250   else { $nichtraucher = 0; }
251
252   if ($abruf_line['alergikerfreundlich'] == 'ja') {
253       $alfreundl = 1;
254   }
255   else { $alfreundl = 0; }
256
257   if ($abruf_line['sportmoeglichkeiten'] == 'ja') {
258       $sport = 1;
259   }
260   else { $sport = 0; }
261
262   if ($abruf_line['wandern'] == 'ja') {
263       $wandern = 1;
264   }
265   else { $wandern = 0; }
266
267   if ($abruf_line['ausstattung-aussenbereich-kinderspielplatz'] == 'ja') {
268       $spielplatz = 1;
269   }
270   else { $spielplatz = 0; }
271
272
```



```

273 $zimmer = $abruf_line['anzahlzimmer_schlafzimmer_kinder']+
274 $abruf_line['anzahlzimmer_wohnzimmer']+
275 $abruf_line['anzahlzimmer_esszimmer']+
276 $abruf_line['anzahlzimmer_wohn_esszimmer']+
277 $abruf_line['anzahlzimmer_kueche'];
278
279 echo 'Objektyp_vor_Merkmaleintr:~';
280
281 $merkmale_eintragen = $strenner2.$idanbieter.$strenner.$id_hotel.new.
282 $strenner.$objektyp.$strenner.''. $strenner.'0'.$strenner.'0'.$strenner.
283 $rollstuhl.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.$garage.
284 $strenner.'0'.$strenner.$klimaanlage.$strenner.'0'.$strenner.$haustiere.
285 $strenner.$kamin.$strenner.$kinderbetreuung.$strenner.$pool.$strenner.
286 '0'.$strenner.$schwimmbad.$strenner.'0'.$strenner.$gesellschaften.
287 $strenner.$personen.$strenner.$kinder.$strenner.$zimmer.$strenner.
288 $schlafzimmer.$strenner.'0'.$strenner.$tv.$strenner.'0'.$strenner.
289 $internet.$strenner.$waschmaschine.$strenner.$herd.$strenner.
290 $kuechlschrank.$strenner.$wellness.$strenner.$whirlpool.$strenner.
291 '0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.
292 '0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.
293 '0'.$strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.$nichtraucher.
294 $strenner.$alfreundl.$strenner.$sport.$strenner.'0'.$strenner.'0'.
295 $strenner.'0'.$strenner.'0'.$strenner.'0'.$strenner.$spielplatz.$strenner2."~\n";
296 fwrite($handle_writefile_merkmale, $merkmale_eintragen);
297
298 /* — Privathäuser Preise, Zusatzleistungen — */
299 // Ist kompliziert, noch nicht verwirklicht
300 $preise = $abruf_line['vermietungstarife'];
301 $preisneu = explode("\n", $preise);
302 //echo $id_hotel_new.':~';
303 //print_r($preisneu);
304 //echo '<br>';
305 //echo $preise.'<br>';
306 //echo $hausbeschreibung_eintragen;
307 //echo $merkmale_eintragen;
308
309 /* — Privathäuser hausbilder — */
310 $verzeichnis2 = $abruf_line['exportverzeichnis'];
311 $idland = $abruf_line['id_land'];
312 $querylanddir = "SELECT_vverzeichnis FROM_a_land WHERE_id_land = '$idland' LIMIT 1";
313 $abruflanddir = mysql_query($querylanddir);
314 while($abruf_line = mysql_fetch_array($abruflanddir, MYSQL_ASSOC)) {
315     $verzeichnis1 = $abruf_line['verzeichnis'];
316 }
317 $url = 'http://www.ferienhaus.de/ferienhaeuser/'. $verzeichnis1.'/'.
318 $verzeichnis2.'/images/';
319
320 $querybilder = "SELECT_id_bild FROM_hotels_bilderliste WHERE
321 id_hotel = '$id_hotel'";
322 $abrufbilder = mysql_query($querybilder);
323 $k = 1;
324 while($abruf_line = mysql_fetch_array($abrufbilder, MYSQL_ASSOC)) {
325     $bild = $abruf_line['id_bild'];
326     $bildlink = $url.$bild.'.jpg';
327
328     //echo 'Bildlink: '. $bildlink.'<br>';
329     if ($verzeichnis2 == $verzeichnis_old) {
330         $k++;
331     }
332     else {
333         $k = 1;
334     }
335     $bilder_eintragen = $strenner2.$idanbieter.$strenner.$id_hotel.new.
336     $strenner.$k.$strenner.'NULL'.$strenner.$bildlink.$strenner2."~\n";
337     fwrite($handle_writefile_bilder, $bilder_eintragen);
338     $verzeichnis_old = $verzeichnis2;

```

```

339
340 }
341
342 /* —— Privathäuser Stammdaten —— */
343 //IDAnbieter, ObjReferenz, Kategorie, Land, Region, Ort
344 $queryland = "SELECT name FROM a_land WHERE id_land = '$id_land' LIMIT 1";
345 $abrufland = mysql_query($queryland);
346 while($abruf_line = mysql_fetch_array($abrufland, MYSQL_ASSOC)) {
347     $landesname = ltrim(rtrim(addslashes(utf8_encode($abruf_line['name']))));
348 }
349
350 $queryregion = "SELECT name FROM a_region WHERE id_region = '$id_region' LIMIT 1";
351 $abrufregion = mysql_query($queryregion);
352 while($abruf_line = mysql_fetch_array($abrufregion, MYSQL_ASSOC)) {
353     $regionsname = ltrim(rtrim(addslashes(utf8_encode($abruf_line['name']))));
354 }
355
356 $stammdaten_eintragen = $strenner2.$idanbieter.$strenner.$id_hotel_new.
357 $strenner.$objekttyp.$strenner.$landesname.$strenner.$regionsname.
358 $strenner.$ortsname.$strenner2."\\n";
359 fwrite($handle_writefile_stammdaten, $stammdaten_eintragen);
360 //echo 'Stammdaten.EINTRAGEN: '. $stammdaten_eintragen. '<br>';
361 $i++;
362
363 /* —— Privathäuser Vakanzen —— */
364 /* Arrays generieren, aktuelles Jahr */
365 // Monate ab jetzt, Tage ab Monatsanfang akt Monat
366 /* init */
367 $vak_jahr = NULL;
368 $vak_jahr2 = NULL;
369 $ueb_jahr = NULL;
370 $ueb_jahr2 = NULL;
371 $mins_jahr = NULL;
372 $mins_jahr2 = NULL;
373
374 for ($i = (int)$monat; $i <= 12; $i++) {
375     $monthdays = date("t", mktime(0, 0, 0, $i, 1, $jahr));
376     for ($k = 1; $k <= $monthdays; $k++) {
377         $vakanzen[$jahr][$i][$k] = 'Y';
378         $uebergabe[$jahr][$i][$k] = 'C';
379         $minstay[$jahr][$i][$k] = 'C';
380     }
381 }
382 // Monate ab 1. des Folgejahres, Tage ab Monatsanfang 1. Monat
383 for ($i = 1; $i <= 12; $i++) {
384     $monthdays = date("t", mktime(0, 0, 0, $i, 1, ($jahr+1)));
385     for ($k = 1; $k <= $monthdays; $k++) {
386         $vakanzen[( $jahr+1)][$i][$k] = 'Y';
387         $uebergabe[( $jahr+1)][$i][$k] = 'C';
388         $minstay[( $jahr+1)][$i][$k] = 'C';
389     }
390 }
391
392 // Vakanzen abfragen
393 $query = "SELECT * FROM stabelle_vak WHERE id_hotel = $id_hotel AND tag >= '$datum'";
394 $result = mysql_query($query);
395 while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
396     $hotelbeleg = $line['tag'];
397     echo $hotelbeleg. '<br>';
398     $datum_vdb = explode("-", $hotelbeleg);
399     $jahr_db = (int)$datum_vdb[0];
400     $monat_db = (int)$datum_vdb[1];
401     $tag_db = (int)$datum_vdb[2];
402
403     //print_r($datum_vdb);
404     $vakanzen[$jahr_db][$monat_db][$tag_db] = 'N';

```

```

405     $uebergabe[$jahr_db][$monat_db][$tag_db] = 'X';
406     $minstay[$jahr_db][$monat_db][$tag_db] = 'C';
407 }
408 //print_r($vakanken);
409 echo $jahr.'<br>';
410 echo $monat.'<br>';
411 echo $tag.'<br>';
412
413 //implodieren der Tage, aktuelles Jahr
414 for ($i = (int)$monat; $i <= 12; $i++) {
415     $vak_string = implode(NULL,$vakanken[$jahr][$i]);
416     $ueb_string = implode(NULL,$uebergabe[$jahr][$i]);
417     $mins_string = implode(NULL,$minstay[$jahr][$i]);
418     $vak_jahr = $vak_jahr.$vak_string;
419     $ueb_jahr = $ueb_jahr.$ueb_string;
420     $mins_jahr = $mins_jahr.$mins_string;
421 }
422
423 //implodieren der Tage, folgendes Jahr
424 for ($i = 1; $i <= 12; $i++) {
425     $vak_string2 = implode(NULL,$vakanken[( $jahr+1)][$i]);
426     $ueb_string2 = implode(NULL,$uebergabe[( $jahr+1)][$i]);
427     $mins_string2 = implode(NULL,$minstay[( $jahr+1)][$i]);
428     $vak_jahr2 = $vak_jahr2.$vak_string2;
429     $ueb_jahr2 = $ueb_jahr2.$ueb_string2;
430     $mins_jahr2 = $mins_jahr2.$mins_string2;
431 }
432
433 //Strings zusammenführen und kürzen
434 $vak_alle = $vak_jahr.$vak_jahr2;
435 $ueb_alle = $ueb_jahr.$ueb_jahr2;
436 $mins_alle = $mins_jahr.$mins_jahr2;
437 $vak_alle = substr($vak_alle, 0, 365);
438 $ueb_alle = substr($ueb_alle, 0, 365);
439 $mins_alle = substr($mins_alle, 0, 365);
440 echo $vak_alle.'<br>';
441 echo $ueb_alle.'<br>';
442 echo $mins_alle.'<br>';
443
444 /* id_hotel -> id_hotel_new */
445 $id_hotel_new = '';
446 $nummernlaenge = strlen($id_hotel);
447 $nullserie = 6-$nummernlaenge;
448 for ($k = 0; $k < $nullserie; $k++) {
449     $id_hotel_new = $id_hotel_new.'0';
450 }
451 $id_hotel_new = 'ONFH'.$id_hotel_new.$id_hotel;
452
453 /* Objekttyp -> Objektkategorie */
454 switch ($objekttyp) {
455     case 'Haus':
456         $objekttyp = 'HA';
457         break;
458     case 'Wohnung':
459         $objekttyp = 'WH';
460         break;
461     case 'Feriendorf':
462         $objekttyp = 'FD';
463         break;
464     case '':
465         $objekttyp = 'KA';
466         break;
467 }
468
469 $eintragen = $strenner2.'99'.$strenner.$id_hotel_new.$strenner.
470 $objekttyp.$strenner.$jahr.'-'. $monat.'-01'.$strenner.$vak_alle.$strenner.

```

```
471     $ueb_alle.$strenner.  
472     $mins_alle.$strenner2."\n";  
473     fwrite($handle_writefile_vakanzen, $eintragen);  
474 }  
475 fclose($handle_writefile_texte);  
476 fclose($handle_writefile_bilder);  
477 fclose($handle_writefile_merkmale);  
478 fclose($handle_writefile_stammdaten);  
479 fclose($handle_writefile_vakanzen);
```

F. Quelltexte der Klassen und Funktionen

F.1. Quelltext der Dataload Klasse

Listing F.1: Quelltext der Dataload Klasse

```
1 class dataload {
2
3     var $dateigroesse;
4     var $eintraege;
5     var $zeilen;
6     var $info_indb;
7     var $info_backupfile;
8     var $info_ftpload;
9     var $landesnamen;
10    var $regionsnamen;
11    var $sortsnamen;
12    var $suchmessage;
13    var $suchswitch;
14    var $land_ref;
15    var $region_ref;
16    var $sort_ref;
17
18
19    function interhome_ftpload($objektverz, $dateiname, $dateinamezip) {
20
21        $pfad = getenv("DOCUMENT_ROOT");
22        $ftp_connect = $pfad.'/include/interhome_ftpconnect.php';
23        include $ftp_connect;
24        if ($ftp_einloggen) {
25            $info = 'erfolgreiches_login_am_Interhome_server, Lade_ZIP_Datei<br>';
26            $datei_laden = ftp_get($ftp_verbinden, $objektverz.$dateinamezip,
27                                $dateinamezip, FTP_BINARY);
28            if ($datei_laden) {
29                $info = $info.'Datei_erfolgreich_geladen<br>';
30            }
31            else {
32                $info = $info.'FEHLER: Datei_konnte_nicht_geladen_werden<br>';
33            }
34        }
35        ftp_close($ftp_verbinden);
36
37        $zip_bilder_appt = new ZipArchive;
38        if ($zip_bilder_appt->open($objektverz.$dateinamezip) == TRUE) {
39
40            $zip_bilder_appt->extractTo($objektverz);
41            $zip_bilder_appt->close();
42            $info = $info.'Datei_erfolgreich_ins_Objektverzeichnis_entpackt<br>';
43
44            if (file_exists($objektverz.$dateinamezip)) {
45                unlink($objektverz.$dateinamezip);
46                $info = $info.'Temporäre_Datei_gelöscht<br>';
```

```

47     }
48 }
49 else {
50     $info = $info.'FEHLER: _Konnte _Datei _nicht _entpacken<br>';
51 }
52 $this->dateigroesse = filesize($objektverz.$dateiname);
53 $this->info_ftpload = $info;
54 }
55
56 function interhome_explodefile($objektverz, $dateiname, $dateigroesse) {
57
58     $handle = fopen($objektverz.$dateiname, "r");
59     $contents = fread($handle, $dateigroesse);
60     $contentneu = strtr($contents, '"', '_');
61     $zeile = explode("\n", $contentneu);
62     $eintraege=count($zeile);
63     fclose($handle);
64
65     $this->zeilen = $zeile;
66     $this->eintraege = $eintraege;
67 }
68
69 function stammdatens_indb($tabelle, $dateiwrite, $idanbieter) {
70     $leeren = "DELETE FROM $tabelle WHERE IDAnbieter = $idanbieter";
71     $loaddbinfile = "LOAD DATA LOCAL INFILE '$dateiwrite' INTO TABLE
72     $tabelle FIELDS TERMINATED BY ';' ENCLOSED BY '\"'";
73     if(!$tblloesch = mysql_query($leeren)) {
74         $info = 'Datensätze konnten nicht gelöscht werden: '. mysql_error();
75     }
76     else {
77         $info = 'Datenbank erfolgreich gelöscht. Fülle mit neuen Daten auf...<br>';
78         if (!$result = mysql_query($loaddbinfile)) {
79             $info = $info.'Daten konnten nicht geschrieben werden: '. mysql_error();
80         }
81         else {
82             $info = $info.'Daten erfolgreich eingelesen';
83         }
84     }
85     $this->info_indb = $info;
86 }
87
88 function interhome_backupfile($objektverz, $backupverz, $dateiname) {
89     echo $objektverz.$dateiname;
90     if (file_exists($objektverz.$dateiname)) {
91         if (file_exists($backupverz.$dateiname)) {
92             unlink($backupverz.$dateiname);
93         }
94         $info = 'Backup alter CSV-Datei...<br>';
95         if (copy($objektverz.$dateiname, $backupverz.$dateiname)) {
96
97             unlink($objektverz.$dateiname);
98             $info = $info.'Erfolgreich von '.$objektverz.$dateiname.' nach '.
99             $backupverz.$dateiname.' kopiert...<br>Dateibackup durchgeführt';
100         }
101         else {
102             $info = $info.'FEHLER, _Datei _nicht _kopiert. _Evtl _Datei _nicht _vorhanden?';
103         }
104     }
105     $this->info_backupfile = $info;
106 }
107
108 function walters_ftpload($objektverz, $dateiname, $dateinamegz) {
109     $pfad = getenv("DOCUMENT_ROOT");
110     $ftp_connect = $pfad.'/include/walters_ftpconnect.php';
111     include $ftp_connect;
112     if ($ftp_einloggen) {

```

```

113     $info = 'erfolgreiches_login_am_Wolters_server ,_Lade_GZ_Datei<br>';
114     $datei_laden = ftp_get($ftp_verbinden,$objektverz.$dateinamegz,
115     $dateinamegz, FTP_BINARY);
116     if ($datei_laden) {
117         $info = $info.'Datei_erfolgreich_geladen<br>';
118     }
119     else {
120         $info = $info.'FEHLER:_Datei_konnte_nicht_geladen_werden<br>';
121     }
122 }
123 ftp_close($ftp_verbinden);
124
125 $dateigroesse = filesize($objektverz.$dateinamegz)*50;
126 $fp = gzopen($objektverz.$dateinamegz, "r");
127 if ($fp) {
128     $info = $info.'Datei_erfolgreich_geöffnet<br>';
129 }
130 else { $info = $info.'FEHLER:_Datei_konnte_nicht_geöffnet_werden<br>'; }
131 $file_contents = gzread($fp, $dateigroesse);
132 if ($file_contents) {
133     $info = $info.'Datei_erfolgreich_gelesen<br>';
134 }
135 else { $info = $info.'FEHLER:_Datei_konnte_nicht_gelesen_werden<br>'; }
136 if (is_writable($dateiname)) {
137     $handle_writefile = fopen($dateiname, "w");
138     fwrite($handle_writefile, $file_contents);
139     $info = $info.'Neue_Datei_erfolgreich_geschrieben<br>';
140 }
141 else { $info = $info.'Datei_'. $dateiname. '
142 nicht_vorhanden_oder_nicht_beschreibbar!<br>'; }
143 gzclose($fp);
144 if (unlink($objektverz.$dateinamegz)) {
145     $info = $info.'Temporäre_Datei_gelöscht<br>';
146 }
147 $this->info_ftpload = $info;
148 }
149
150 function wolters_indb($tabelle, $dateiwrite) {
151     $leeren = "DELETE FROM $tabelle WHERE ID_Anbieter = 2";
152     $loaddbinfile = "LOAD DATA LOCAL INFILE '$dateiwrite' INTO TABLE
153     $tabelle FIELDS TERMINATED BY ';' ENCLOSED BY '\"' ";
154     if (!$tblloesch = mysql_query($leeren)) {
155         $info = 'Datensätze_konnten_nicht_gelöscht_werden:'. mysql_error();
156     }
157     else {
158         $info = 'Datenbank_erfolgreich_gelöscht._Fülle_mit_neuen_Daten_auf...<br>';
159         if (!$result = mysql_query($loaddbinfile)) {
160             $info = $info.'Daten_konnten_nicht_geschrieben_werden:'. mysql_error();
161         }
162         else {
163             $info = $info.'Daten_erfolgreich_eingelesen';
164         }
165     }
166     $this->info_indb = $info;
167 }
168
169 function onferienhaus_ftpload($objektverz, $dateiname) {
170
171     $pfad = getenv("DOCUMENT_ROOT");
172     $ftp_connect = $pfad.'/include/onferienhaus_ftpconnect.php';
173     include $ftp_connect;
174     if ($ftp_einloggen) {
175         $info = 'erfolgreiches_login_am_Online_Now_Ferienhaus_Server ,
176         _Lade_Datei<br>';
177         $datei_laden = ftp_get($ftp_verbinden,$objektverz.$dateiname,
178         '/html/export/'.$dateiname, FTP_BINARY);

```

```

179     if ($datei_laden) {
180         $info = $info.'Datei_erfolgreich_geladen<br>';
181     }
182     else {
183         $info = $info.'FEHLER: _Datei_konnte_nicht_geladen_werden<br>';
184     }
185 }
186 ftp_close($ftp_verbinden);
187
188 /*$zip_bilder_appt = new ZipArchive;
189 if ($zip_bilder_appt->open($objektverz.$dateinamezip) === TRUE) {
190
191     $zip_bilder_appt->extractTo($objektverz);
192     $zip_bilder_appt->close();
193     $info = $info.'Datei_erfolgreich_ins_Objektverzeichnis_entpackt<br>';
194
195     if (file_exists($objektverz.$dateinamezip)) {
196         unlink($objektverz.$dateinamezip);
197         $info = $info.'Temporäre Datei gelöscht<br>';
198     }
199 }
200 else {
201     $info = $info.'FEHLER: Konnte Datei nicht entpacken<br>';
202 } */
203 $this->dateigroesse = filesize($objektverz.$dateiname);
204 $this->info_ftpload = $info;
205 }
206
207 function onferienhaus_indb($tabelle, $dateiwrite) {
208     $leeren = "DELETE FROM $tabelle WHERE ID_Anbieter = 99";
209     $loaddbinfile = "LOAD DATA LOCAL INFILE '$dateiwrite' INTO TABLE
210     $tabelle FIELDS TERMINATED BY ';' ENCLOSED BY '\"'";
211     if (!$tblloesch = mysql_query($leeren)) {
212         $info = 'Datensätze konnten nicht gelöscht werden: '. mysql_error();
213     }
214     else {
215         $info = 'Datenbank erfolgreich gelöscht. _Fülle_mit_neuen_Daten_auf...<br>';
216         if (!$result = mysql_query($loaddbinfile)) {
217             $info = $info.'Daten konnten nicht geschrieben werden: '. mysql_error();
218         }
219         else {
220             $info = $info.'Daten erfolgreich eingelesen';
221         }
222     }
223     $this->info_indb = $info;
224 }
225
226
227
228
229 function geo_zuweisung($objektverz, $dateiname, $typ) {
230     $i = 0;
231     $k = 0;
232     $dateigroesse = filesize($objektverz.$dateiname);
233     $handle = fopen($objektverz.$dateiname, "r");
234     $contents = fread($handle, $dateigroesse);
235     $contentneu = strtr($contents, '"', '_');
236     $zeile = explode("\n", $contentneu);
237     $eintraege = count($zeile);
238     echo $eintraege.'Eingetragene Zeilen_(geo_zuweisung)<br>';
239
240     foreach($zeile AS $blabla) {
241         $spaltenwerte = explode(";", $zeile[$i]);
242         // $eintraege = count($spaltenwerte);
243         $i++;
244         if ($typ == 'land') {

```



```

245     $sprachcode = substr($spaltenwerte[4], 0, 2);
246     if (preg_match("/\bde\b/i", $sprachcode)) {
247         $geo_array[$k] = $spaltenwerte;
248         $k++;
249     }
250 }
251 if ($typ == 'region') {
252     $sprachcode = substr($spaltenwerte[3], 0, 2);
253     if (preg_match("/\bde\b/i", $sprachcode)) {
254         $geo_array[$k] = $spaltenwerte;
255         $k++;
256     }
257 }
258 if ($typ == 'ort') {
259     $geo_array[$k] = $spaltenwerte;
260     $k++;
261 }
262
263
264 }
265 if ($typ == 'land') {
266     $this->landesnamen = $geo_array;
267 }
268 if ($typ == 'region') {
269     $this->regionsnamen = $geo_array;
270 }
271 if ($typ == 'ort') {
272     $this->ortsnamen = $geo_array;
273 }
274 }
275
276 function stammdaten_landesvergleich($fh_land_name, $fh_region_name, $objstadt) {
277     $schalter = 0;
278     $message = '<span_style="color:#444444;">----Landessuche----</span><br>';
279     $schalter = 0;
280     $query_fh_land = "SELECT_*_FROM_ref_land_WHERE_Land_LIKE_'" . $fh_land_name . "'";
281     $aufruf_fh_land = mysql_query($query_fh_land);
282     $zeilen_fh_land = mysql_num_rows($aufruf_fh_land);
283     if ($zeilen_fh_land > 1) {
284         /* ----- Start Genauere Suche nach dem Land ----- */
285         $message = $message . '<span_style="color:#999900;">es wurden mehrere
286         mögliche_Länder_ausgemacht_Suche_genauer</span><br>';
287         $query_fh_land = "SELECT_*_FROM_ref_land_WHERE_Land_LIKE_'" . $fh_land_name . "'";
288         $aufruf_fh_land = mysql_query($query_fh_land);
289         $zeilen_fh_land = mysql_num_rows($aufruf_fh_land);
290         if ($zeilen_fh_land > 1) {
291             $message = $message . '<span_style="color:#DD0000;">
292             Warnung, fh_land hat Mehrfach identische Einträge.
293             Nehme letzten Eintrag aus der Liste und weise zu (genaue Suche)
294             </span><br>';
295             while ($line_fh_land = mysql_fetch_array($aufruf_fh_land)) {
296                 $land_last = $line_fh_land['IDLand'];
297                 $message = $message . '<span_style="color:#999999;">Länder: _'.
298                 $line_fh_land['Land']. '_Kürzel: _'. $line_fh_land['IDLand']. '</span><br>';
299             }
300             $this->land_ref = $land_last;
301             //$this->region_ref = $region_last;
302
303             $schalter = 'regionsuchen';
304             //include 'interhome_stamm_regionssuche.php';
305         }
306     elseif ($zeilen_fh_land == 1) {
307         /* ----- LandID holen (in genaue Suche) ----- */
308         $message = $message . '<span_style="color:#009900;">
309         Genau einen Eintrag gefunden_Auflistung_(genaue_Suche):</span><br>';
310         while ($line_fh_land = mysql_fetch_array($aufruf_fh_land)) {

```

```

311         $land_last = $line_fh_land['IDLand'];
312         $message = $message.'<span style="color:#999999;">Land:␣'.
313         $line_fh_land['Land']. 'Kürzel:␣'. $line_fh_land2['IDLand']. '</span><br>';
314     }
315     $this->land_ref = $land_last;
316     // $this->region_ref = $region_last;
317
318     $schalter = 'regionsuchen';
319     //include 'interhome_stamm_regionssuche.php';
320 }
321 elseif ($zeilen_fh_land < 1) {
322     /* -Kein Land vorhanden in fh_land, neues eintragen (in genaue Suche)- */
323     $message = $message.'<span style="color:#999900;">
324     Keinen Eintrag in der Landestabelle gefunden, füge neues Land in die
325     Tabelle (genaue Suche):</span><br>';
326     $land_last = $land;
327     $message = $message.'<span style="color:#999999;">LandLast:␣'.
328     $land_last.'Land:␣'. $fh_land_name.'</span><br>';
329     $insert_fh_land = "INSERT INTO ref_land (IDLand, Land_kurz, Land)
330     VALUES ('$land_last', '$land_last', '$fh_land_name')";
331     if (mysql_query($insert_fh_land)) {
332         $message = $message.'<span style="color:#009900;">Erfolgreiches INSERT
333         in fh_land</span><br>';
334     }
335     else {
336         $message = $message.'<span style="color:#DD0000;">FEHLER,
337         Datensatz nicht eingefügt in fh_land'.mysql_error(). '</span><br>';
338     }
339     //Ermittlung der letzten RegionsID
340     $query_regionstabelle = "SELECT IDRegion FROM ref_region ORDER BY IDRegion";
341     $aufruf_regionstabelle = mysql_query($query_regionstabelle);
342     while ($line_aufruf_region = mysql_fetch_array($aufruf_regionstabelle)) {
343         $regionsid = $line_aufruf_region['IDRegion'];
344     }
345     $region_last = substr($regionsid,0,2);
346     $message = $message.'<span style="color:#999999;">Alte Regionsid-2 Stellen'.
347     $region_last.'</span><br>';
348     $region_last = $region_last+1;
349     $region_last = $region_last.'01';
350     $regionort1 = $region_last.'001';
351     $ort_last = $regionort1;
352     //Die erste Region des Landes in fh_region einfügen
353     $insert_fh_landregion1 = "INSERT INTO ref_region (IDRegion, LandID, Region)
354     VALUES ('$region_last', '$land_last', '$fh_region_name')";
355     if (mysql_query($insert_fh_landregion1)) {
356         $message = $message.'<span style="color:#009900;">
357         Erfolgreiches INSERT der 1. Region'. $fh_region_name.' mit der ID:␣'.
358         $region_last.' in fh_ort</span><br>';
359     }
360     }
361     else {
362         $message = $message.'<span style="color:#DD0000;">
363         FEHLER, Datensatz nicht eingefügt in fh_ort'.mysql_error(). '</span><br>';
364     }
365     //Den ersten Ort der Region in fh_ort einfügen
366     $insert_fh_regionort1 = "INSERT INTO ref_ort
367     (IDOrt, RegionID, LandID, Skiort, Ort) VALUES ('$regionort1', '$region_last',
368     '$land_last', '0', '$objstadt')";
369     if (mysql_query($insert_fh_regionort1)) {
370         $message = $message.'<span style="color:#009900;">
371         Erfolgreiches INSERT des 1. Ortes'. $objstadt.' mit der ID:␣'.
372         $ort_last.' in fh_ort</span><br>';
373     }
374     else {
375         $message = $message.'<span style="color:#DD0000;">
376         FEHLER, Datensatz nicht eingefügt in fh_ort'.mysql_error().

```

```

377         '</span><br>';
378     }
379 }
380 /* — Ende Genauere Suche nach dem Land — */
381 }
382 elseif ($zeilen_fh_land == 1) {
383     /* — RegionID holen — */
384     $message = $message. '<span style="color:#009900;">
385     Genau_einen_Eintrag_gefunden_Auflistung:<br>';
386     while ($line_fh_land = mysql_fetch_array($aufruf_fh_land)) {
387         $land_last = $line_fh_land['IDLand'];
388         $message = $message. '<span style="color:#999999;">Land: '.
389         $line_fh_land['Land']. 'Kürzel: '. $line_fh_land['IDLand']. '</span><br>';
390     }
391     $this->land_ref = $land_last;
392     //$this->region_ref = $region_last;
393
394     $schalter = 'regionsuchen';
395     //include 'interhome_stamm-regionssuche.php';
396 }
397 elseif ($zeilen_fh_land < 1) {
398     /* — Kein Land vorhanden in fh_land, neues Land eintragen — */
399     $message = $message. '<span style="color:#999900;">
400     Keinen_Eintrag_in_der_Landestabelle_gefunden, füge neues Land in
401     die_Tabelle:</span><br>';
402     $land_last = $land;
403     $message = $message. '<span style="color:#999999;">LandLast: '.
404     $land_last. 'Land: '. $fh_land_name. '</span><br>';
405     $insert_fh_land = "INSERT INTO ref_land (IDLand, Land_kurz, Land)
406     VALUES (' $land_last ', ' $land_last ', ' $fh_land_name ')";
407     if (mysql_query($insert_fh_land)) {
408         $message = $message. '<span style="color:#009900;">
409         Erfolgreiches INSERT in fh_land</span><br>';
410     }
411     else {
412         $message = $message. '<span style="color:#DD0000;">
413         FEHLER, Datensatz nicht eingefügt in fh_land'. mysql_error(). '</span><br>';
414     }
415     //Ermittlung der letzten RegionsID
416     $query_regionstabelle = "SELECT IDRegion FROM ref_region ORDER BY IDRegion";
417     $aufruf_regionstabelle = mysql_query($query_regionstabelle);
418     while ($line_aufruf_region = mysql_fetch_array($aufruf_regionstabelle)) {
419         $regionsid = $line_aufruf_region['IDRegion'];
420     }
421     $region_last = substr($regionsid,0,2);
422     //echo 'Alte Regionsid-2 Stellen '. $region_last. '<br>';
423     $region_last = $region_last+1;
424     $region_last = $region_last.'01';
425     $regionort1 = $region_last.'001';
426     $ort_last = $regionort1;
427     //Die erste Region des Landes in fh_region einfügen
428     $insert_fh_landregion1 = "INSERT INTO ref_region (IDRegion, LandID, Region)
429     VALUES (' $region_last ', ' $land_last ', ' $fh_region_name ')";
430     if (mysql_query($insert_fh_landregion1)) {
431         $message = $message. '<span style="color:#009900;">
432         Erfolgreiches INSERT der 1. Region'. $fh_region_name. ' mit der ID: '.
433         $region_last. ' in fh_ort</span><br>';
434     }
435     else {
436         $message = $message. '<span style="color:#DD0000;">
437         FEHLER, Datensatz nicht eingefügt in fh_region ... Region'.
438         $fh_region_name. ' mit der ID: '. $region_last. ' in fh_ort'.
439         mysql_error(). '</span><br>';
440     }
441     //Den ersten Ort der Region in fh_ort einfügen
442     $insert_fh_regionort1 = "INSERT INTO ref_ort

```

```

443 (IDort, RegionID, LandID, Skiort, Ort) VALUES ('$regionort1', '$region_last',
444 '$land_last', '0', '$objstadt');
445 if (mysql_query($insert_fh_regionort1)) {
446     $message = $message. '<span style="color:#009900;">
447     Erfolgreiches INSERT des 1. Ortes: '. $objstadt. ' mit der ID: '.
448     $ort_last. ' in fh_ort</span><br>';
449 }
450 else {
451     $message = $message. '<span style="color:#DD0000;">
452     FEHLER, Datensatz nicht eingefügt in fh_ort ... Ort: '.
453     $objstadt. ' mit der ID: '. $ort_last. ' '. mysql_error(). '</span><br>';
454 }
455 }
456 $this->suchswitch = $schalter;
457 $this->suchmessage = $message;
458 }
459
460 function stammdaten_regionsvergleich($land_ref, $fh_region_name, $objstadt) {
461     // $objstadt fehlt noch
462     $message = '<span style="color:#444444;"> Interhome Regionssuche</span><br>';
463
464     $schalter = 0;
465     $query_fh_region = "SELECT * FROM ref_region WHERE LandID = '$land_ref' AND
466     Region LIKE '%$fh_region_name'";
467     $aufruf_fh_region = mysql_query($query_fh_region);
468     $zeilen_fh_region = mysql_num_rows($aufruf_fh_region);
469     if ($zeilen_fh_region > 1) {
470         /* — Anfang Genauere Suche nach der Region — */
471         $message = $message. '<span style="color:#FF0000;">
472         es wurden mehrere mögliche Regionen ausgemacht Suche genauer</span><br>';
473         $query_fh_region = "SELECT * FROM ref_region WHERE LandID = '$land_ref' AND
474         Region LIKE '$fh_region_name'";
475         $aufruf_fh_region = mysql_query($query_fh_region);
476         $zeilen_fh_region = mysql_num_rows($aufruf_fh_region);
477         if ($zeilen_fh_region > 1) {
478             $message = $message. '<span style="color:#FF0000;">
479             Warnung, fh_region hat Mehrfach identische Einträge.
480             Nehme letzten Eintrag aus der Liste und weise zu (in genaue Suche)
481             </span><br>';
482             while ($line_fh_region = mysql_fetch_array($aufruf_fh_region)) {
483                 $region_last = $line_fh_region['IDRegion'];
484                 $message = $message. '<span style="color:#999999;"> Regionen: '.
485                 $line_fh_region['Region']. ' Nummer: '. $line_fh_region['IDRegion'].
486                 '</span><br>';
487             }
488             $schalter = 'ortsuchen';
489             $this->region_ref = $region_last;
490             /* $dataload->stammdaten_ortsvergleich($land_ref, $region_ref, $objstadt); */
491             //include 'interhome_stamm_ortssuche.php';
492         }
493     } elseif ($zeilen_fh_region == 1) {
494         /* — RegionID holen (in genaue Suche) — */
495         $message = $message. '<span style="color:#00FF00;">
496         Genau einen Eintrag gefunden Auflistung (in genaue Suche):</span><br>';
497         while ($line_fh_region = mysql_fetch_array($aufruf_fh_region)) {
498             $region_last = $line_fh_region['IDRegion'];
499             $message = $message. '<span style="color:#999999;"> Region: '.
500             $line_fh_region['Region']. ' Nummer: '. $line_fh_region['IDRegion'].
501             '</span><br>';
502         }
503         $schalter = 'ortsuchen';
504         $this->region_ref = $region_last;
505         /* $dataload->stammdaten_ortsvergleich($land_ref, $region_ref, $objstadt); */
506         //include 'interhome_stamm_ortssuche.php';
507     }
508     elseif ($zeilen_fh_region < 1) {

```

```

509      /* -Keine Region vorhanden in fh_region, neue eintragen (in genaue Suche)- */
510      $message = $message.'<span style="color:#999900;">
511      Keinen Eintrag in der Regionstabelle gefunden,
512      füge neue Region in die Tabelle (in genaue Suche):</span><br>';
513      $query_fh_region_neu = "SELECT * FROM ref_region WHERE
514      LandID = '$land_ref' ORDER BY IDRegion";
515      $aufruf_fh_region_neu = mysql_query($query_fh_region_neu);
516
517      $regionen = mysql_num_rows($aufruf_fh_region_neu);
518      $neueregionen = 0;
519      if ($regionen < 1) {
520          $neueregionen = 1;
521          $query_fh_region_neu = "SELECT IDRegion FROM fh_region ORDER BY IDRegion";
522          $aufruf_fh_region_neu = mysql_query($query_fh_region_neu);
523      }
524
525      while ($line_aufruf_fh_region = mysql_fetch_array($aufruf_fh_region_neu,
526      MYSQL_ASSOC)) {
527          $region_last = $line_aufruf_fh_region['IDRegion'];
528          //echo 'ORT: '. $line_aufruf_fh_ort['IDOrt']. '<br>';
529      }
530      $message = $message.'<span style="color:#999999;"> Ermittelte RegionID: '.
531      $region_last.' LandID: '. $land_ref.'</span><br>';
532
533      if ($neueregionen == 1) {
534          $region_last = substr($region_last, 0, 2);
535          $region_last = $region_last+1;
536          $region_last = $region_last.'01';
537      }
538      else {
539          $region_last = $region_last+1; //Da kein neues Land, sondern nur neue Region
540          \\werden die ersten beiden Stellen beibehalten (simples Hochzählen
541          ausgehend von letzter Regionsid dieses Landes)
542      }
543
544      $message = $message.'<span style="color:#999999;"> RegionLast+1: '.
545      $region_last.' Land: '. $land_ref.' Region: '. $fh_region_name.'</span><br>';
546      $insert_fh_region = "INSERT INTO ref_region (IDRegion, LandID, Region) VALUES
547      ('$region_last', '$land_ref', '$fh_region_name')";
548      if (mysql_query($insert_fh_region)) {
549          $message = $message.'<span style="color:#00FF00;">
550      Erfolgreiches INSERT in fh_region</span><br>';
551      }
552      else {
553          $message = $message.'<span style="color:#FF0000;">
554      FEHLER, Datensatz nicht eingefügt in fh_region'. mysql_error().
555      '</span><br>';
556      }
557      //Den ersten Ort der Region in fh_ort einfügen
558      $ort_last = $region_last.'001';
559      $insert_fh_regionort1 = "INSERT INTO ref_ort
560      (IDOrt, RegionID, LandID, Skiort, Ort) VALUES ('$regionort1', '$region_last',
561      '$land', '0', '$objstadt')";
562      if (mysql_query($insert_fh_regionort1)) {
563          $message = $message.'<span style="color:#00FF00;">
564      Erfolgreiches INSERT des 1. Ortes: '. $objstadt.' mit der ID: '.
565      $ort_last.' in fh_ort</span><br>';
566      }
567      else {
568          $message = $message.'<span style="color:#FF0000;">
569      FEHLER, Datensatz nicht eingefügt in fh_ort ... Ort: '.
570      $objstadt.' mit der ID: '. $ort_last.''. mysql_error(). '</span><br>';
571      }
572      }
573      /* ----- Ende Genauere Suche nach der Region ----- */
574      }

```

```

575     elseif ($zeilen_fh_region == 1) {
576         /* ——— RegionID holen ——— */
577         $message = $message.'<span style="color:#009900;">
578         Genau einen Eintrag gefunden Auflistung:</span><br>';
579         while ($line_fh_region = mysql_fetch_array($aufruf_fh_region)) {
580             $region_last = $line_fh_region['IDRegion'];
581             $message = $message.'<span style="color:#999999;">Region:'.
582             $line_fh_region['Region'].'_Nummer:'. $line_fh_region['IDRegion'].
583             '</span><br>';
584         }
585         $schalter = 'ortsuchen';
586         $this->region_ref = $region_last;
587         /*$dataloader->stammdaten_ortsvergleich($land_ref, $region_ref, $objstadt); */
588         //include 'interhome_stamm_ortssuche.php';
589     }
590     elseif ($zeilen_fh_region < 1) {
591         /* ——— Keine Region vorhanden in fh-Region, neue Region eintragen ——— */
592         $message = $message.'<span style="color:#999900;">
593         Keinen Eintrag in der Regionstabelle gefunden,
594         füge neue Region in die Tabelle:</span><br>';
595         $query_fh_region_neu = "SELECT_*_FROM_ref_region WHERE
596         LandID= '$land_ref' ORDER BY IDRegion";
597         $aufruf_fh_region_neu = mysql_query($query_fh_region_neu);
598         $regionen = mysql_num_rows($aufruf_fh_region_neu);
599         $neuregionen = 0;
600         if ($regionen < 1) {
601             $neuregionen = 1;
602             $query_fh_region_neu = "SELECT IDRegion FROM ref_region ORDER BY IDRegion";
603             $aufruf_fh_region_neu = mysql_query($query_fh_region_neu);
604         }
605
606         while ($line_aufruf_fh_region = mysql_fetch_array($aufruf_fh_region_neu,
607             MYSQL_ASSOC)) {
608             $region_last = $line_aufruf_fh_region['IDRegion'];
609             //echo 'ORT: '. $line_aufruf_fh_ort['IDOrt'].'<br>';
610             //$message = $message.'<span style="color:#999999;">'.
611             $region_last.'LandID:'. $landID.'</span><br>';
612         }
613         $message = $message.'<span style="color:#999999;">Ermittelte RegionID:'.
614         $region_last.'LandID:'. $land_ref.'</span><br>';
615
616         if ($neuregionen == 1) {
617             $region_last = substr($region_last,0,2);
618             $region_last = $region_last+1;
619             $region_last = $region_last.'01';
620         }
621         else {
622             $region_last = $region_last+1; //Da kein neues Land, sondern nur neue
623             \\Region werden die ersten beiden Stellen beibehalten
624             \\(simples Hochzählen ausgehend von letzter Regionsid dieses Landes)
625         }
626
627         $message = $message.'<span style="color:#999999;">RegionLast+1:'.
628         $region_last.'Land:'. $land_ref.'_Region:'. $fh_region_name.'</span><br>';
629         $insert_fh_region = "INSERT INTO ref_region (IDRegion, LandID, Region)
630         VALUES ('$region_last', '$land_ref', '$fh_region_name')";
631         if (mysql_query($insert_fh_region)) {
632             $message = $message.'<span style="color:#00FF00;">
633             Erfolgreiches INSERT in fh-region</span><br>';
634         }
635         else {
636             $message = $message.'<span style="color:#FF0000;">
637             FEHLER, Datensatz nicht eingefügt in fh-region'. mysql_error().
638             '</span><br>';
639         }
640         //Den ersten Ort der Region in fh_ort einfügen

```

```

641     $regionort1 = $region_last.'001';
642     $ort_last = $regionort1;
643     $insert_fh_regionort1 = "INSERT INTO ref_ort
644     (IDort, RegionID, LandID, Skioort, Ort) VALUES (' $regionort1 ', ' $region_last ',
645     '$land_ref ', '0', '$objstadt ')" ;
646     if (mysql_query($insert_fh_regionort1)) {
647         $message = $message.'<span style="color:#00FF00;">
648     Erfolgreiches INSERT des 1. Ortes: '. $objstadt.' mit der ID: '.
649     $ort_last.' in fh_ort</span><br>';
650     }
651     else {
652         $message = $message.'<span style="color:#FF0000;">
653     FEHLER, Datensatz nicht eingefügt in fh_ort ... Ort: '.
654     $objstadt.' mit der ID: '. $ort_last.''.mysql_error(). '</span><br>';
655     }
656     $this->region_ref = $region_last;
657     $this->ort_ref = $ort_last;
658 }
659 $this->suchswitch = $schalter;
660 $this->suchmessage = $message;
661 }
662
663 function stammdaten_ortsvergleich($land_ref, $region_ref, $ort) {
664     $message = '<span style="color:#444444;">Interhome Ortssuche</span><br>';
665     $query_fh_ort = "SELECT * FROM ref_ort WHERE LandID = '$land_ref'
666     AND RegionID = '$region_ref' AND Ort LIKE '%$ort' ORDER BY IDort";
667     $aufruf_fh_ort = mysql_query($query_fh_ort);
668     $zeilen_fh_ort = mysql_num_rows($aufruf_fh_ort);
669     if ($zeilen_fh_ort > 1) {
670         /* — Anfang Genauere Suche nach dem Ort — */
671         $message = $message.'<span style="color:#FF0000;">
672     Es wurden mehrere mögliche Orte ausgemacht. Suche genauer, Ort heißt: - '.
673     $ort.'-</span><br>';
674         $query_fh_ort = "SELECT * FROM ref_ort WHERE LandID = '$land_ref'
675     AND RegionID = '$region_ref' AND Ort LIKE '$ort' ORDER BY IDort";
676         $aufruf_fh_ort = mysql_query($query_fh_ort);
677         $zeilen_fh_ort = mysql_num_rows($aufruf_fh_ort);
678         if ($zeilen_fh_ort > 1) {
679             $message = $message.'<span style="color:#FF0000;">
680     Warnung, fh_ort hat Mehrfach identische Einträge. Nehme letzten Eintrag
681     aus der Liste und weise zu (in genaue Suche)</span><br>';
682             while ($line_fh_ort = mysql_fetch_array($aufruf_fh_ort)) {
683                 $ort_last = $line_fh_ort['IDort'];
684                 $message = $message.'<span style="color:#999999;">
685     in func fh_ort gr 1 Orte: '. $line_fh_ort['Ort']. 'Nummer: '.
686     $line_fh_ort['IDort']. '</span><br>';
687             }
688             $this->ort_ref = $ort_last;
689         }
690     elseif ($zeilen_fh_ort == 1) {
691         /* — OrtID holen (in genaue Suche) — */
692         $message = $message.'<span style="color:#00CC00;">
693     Genau einen Eintrag gefunden. Auflistung (in genaue Suche):</span><br>';
694         while ($line_fh_ort = mysql_fetch_array($aufruf_fh_ort)) {
695             $ort_last = $line_fh_ort['IDort'];
696             echo 'Orte: '. $line_fh_ort['Ort']. 'Nummer: '. $line_fh_ort['IDort']. '<br>';
697         }
698         $this->ort_ref = $ort_last;
699     }
700     elseif ($zeilen_fh_ort < 1) {
701         /* — Kein Ort vorhanden in fh_ort, neuen eintragen (in genaue Suche) — */
702         $message = $message.'<span style="color:#BBBB00;">
703     Keinen Eintrag in der Ortstabelle gefunden, füge neuen Ort in die Tabelle
704     (in genaue Suche):</span><br>';
705         $query_fh_ort_neu = "SELECT * FROM ref_ort WHERE LandID = '$land_ref'
706     AND RegionID = '$region_ref' ORDER BY IDort";

```

```

707     $aufruf_fh_ort_neu = mysql_query($query_fh_ort_neu);
708     while ($line_aufruf_fh_ort = mysql_fetch_array($aufruf_fh_ort_neu ,
709     MYSQL_ASSOC)) {
710         $ort_last = $line_aufruf_fh_ort['IDOrt'];
711         //echo 'ORT: '. $line_aufruf_fh_ort['IDOrt']. '<br>';
712     }
713     $ort_last = $ort_last+1;
714     if ($ort_last < 1000000) {
715         $ort_last = $region_ref.'001';
716     }
717
718     $message = $message.'<span_style="color:#444444;">Last+1: '.
719     $ort_last.'Region: '. $region_ref.'Land: '. $land_ref.'Stadt: '.
720     $ort.'</span><br>';
721     $insert_fh_ort = "INSERT INTO ref_ort (IDOrt,RegionID,LandID,Skiort,Ort)
722     VALUES(' $ort_last ',' $region_ref ',' $land_ref ',' 0 ',' $ort )";
723     if (mysql_query($insert_fh_ort)) {
724         $message = $message.'<span_style="color:#00CC00;">
725     Erfolgreiches INSERT in fh_ort </span><br>';
726     }
727     else {
728         $message = $message.'<span_style="color:#FF0000;">
729     FEHLER, Datensatz nicht eingefügt in fh_ort'.mysql_error()).'</span><br>';
730     }
731     $this->ort_ref = $ort_last;
732 }
733 }
734 /* — Ende Genauere Suche nach dem Ort — */
735
736 /* — Wenn nicht genauer gesucht werden musste — */
737 elseif ($zeilen_fh_ort == 1) {
738     /* — OrtID holen — */
739     $message = $message.'<span_style="color:#00CC00;">
740     Genau einen Eintrag gefunden Auflistung:</span><br>';
741     while ($line_fh_ort = mysql_fetch_array($aufruf_fh_ort)) {
742         $ort_last = $line_fh_ort['IDOrt'];
743         $message = $message.'<span_style="color:#999999;">Orte: '.
744         $line_fh_ort['Ort']. 'Nummer: '. $line_fh_ort['IDOrt']. '</span><br>';
745     }
746     $this->ort_ref = $ort_last;
747 }
748 elseif ($zeilen_fh_ort < 1) {
749     /* — Kein Ort vorhanden in fh_ort, neuen Ort eintragen — */
750     $message = $message.'<span_style="color:#BBBB00;">
751     Keinen Eintrag in der Ortstabelle gefunden,
752     füge neuen Ort in die Tabelle:</span><br>';
753     $query_fh_ort_neu = "SELECT * FROM ref_ort WHERE LandID = ' $land_ref '
754     AND RegionID = $region_ref ORDER BY IDOrt";
755     $aufruf_fh_ort_neu = mysql_query($query_fh_ort_neu);
756     while ($line_aufruf_fh_ort = mysql_fetch_array($aufruf_fh_ort_neu ,
757     MYSQL_ASSOC)) {
758         $ort_last = $line_aufruf_fh_ort['IDOrt'];
759         //echo 'ORT: '. $line_aufruf_fh_ort['IDOrt']. '<br>';
760     }
761     $ort_last = $ort_last+1;
762     if ($ort_last < 1000000) {
763         $ort_last = $region_ref.'001';
764     }
765
766     $message = $message.'<span_style="color:#999999;">Last+1: '. $ort_last .
767     'Region: '. $region_ref.'Land: '. $land_ref.'Stadt: '. $ort.'</span><br>';
768     $insert_fh_ort = "INSERT INTO ref_ort (IDOrt,RegionID,LandID,Skiort,Ort)
769     VALUES(' $ort_last ',' $region_ref ',' $land_ref ',' 0 ',' $ort )";
770     if (mysql_query($insert_fh_ort)) {
771         $message = $message.'<span_style="color:#00CC00;">
772     Erfolgreiches INSERT in fh_ort </span><br>';

```



```
773     }
774     else {
775         $message = $message.'<span style="color:#FF0000;">
776         FEHLER, Datensatz nicht eingefügt in fh_ort '.mysql_error(). '</span><br>';
777     }
778     $this->ort_ref = $ort_last;
779 }
780 $this->suchswitch = 0;
781 $this->suchmessage = $message;
782 }
783
784 }
```

F.2. Quelltext der Zeitfunktionen Klasse

Listing F.2: Quelltext der Zeitfunktionen Klasse

```

1 class zeitfunktionen {
2
3     var $datum_iso;
4     var $datum_unix;
5     var $datum_de;
6     var $daysamon;
7     var $daysnmon;
8     var $nextmonth;
9     var $startdamon;
10    var $enddamon;
11    var $startdnmon;
12    var $enddnmon;
13    var $tblkopfmon;
14
15    /* Erwartet ein Datum im Format TT.MM.YYYY */
16    function datum_deisoun($datum_de) {
17        $datum_array = explode(".", $datum_de);
18        $this->datum_iso = $datum_array[2].'-'. $datum_array[1].'-'. $datum_array[0];
19        $this->datum_unix = mktime(0, 0, 0, $datum_array[1],
20            $datum_array[0], $datum_array[2]);
21    }
22 }
23
24    /* Erwartet ein Datum im Format YYYY-MM-TT */
25    function datum_isode($datum_iso) {
26        $datum_array = explode("-", $datum_iso);
27        $this->datum_de = $datum_array[2].'-'. $datum_array[1].'-'. $datum_array[0];
28    }
29
30    /* benötigt ein Array in der Form 0=>DD 1=>MM 2=>YYYY */
31    function days_act_month($datum) {
32        $this->daysamon = date("t", mktime(0, 0, 0, $datum[1], $datum[0], $datum[2]));
33    }
34
35    /* benötigt ein Array in der Form 0=>DD 1=>MM 2=>YYYY */
36    function days_next_month($datum) {
37        if ($datum[0] > 28) {
38            $datum[0] = 28; //sonst springt mktime in den Folgemonat,
39                //hier ausgehend vom Februar
40        }
41        if ($datum[1] == 12) { //Jahreswechsel berücksichtigen
42            $this->daysnmon = date("t", mktime(0, 0, 0, $datum[1]-11,
43                $datum[0], $datum[2]+1));
44        }
45        else {
46            $this->daysnmon = date("t", mktime(0, 0, 0, $datum[1]+1, $datum[0], $datum[2]));
47        }
48    }
49
50    /* erwartet Monat im Format MM */
51    function thistonextmonth($monat) {
52        if ($monat < 9) { //9 deshalb, da ja um 1mon hochgezählt wird
53            //und sich bei 9 => 10 ergibt
54            $datum_monat_vor = $monat+1;
55            $this->nextmonth = '0'. $datum_monat_vor;
56        }
57        elseif ($monat == 12) {
58            $this->nextmonth = '01';
59        }
60        else {
61            $this->nextmonth = $monat+1;

```

```

62     }
63 }
64
65 function startendactmonth($month, $monatstage) {
66     $this->startdamon = $month[2].'-'. $month[1].'-01';
67     $this->enddamon = $month[2].'-'. $month[1].'-'. $monatstage;
68 }
69
70 function startendnexmonth($month, $monatstage, $nextmonth) {
71     if ($month[1] == 12) {
72         $this->startdnmon = 1+$month[2].'-01-01';
73         $this->enddnmon = 1+$month[2].'-01-'. $monatstage;
74     }
75     else {
76         //Folgemonat festlegen und bei Bedarf eine Null anhängen
77         $this->startdnmon = $month[2].'-'. $nextmonth.'-01';
78         $this->enddnmon = $month[2].'-'. $nextmonth.'-'. $monatstage;
79     }
80 }
81
82 function tabelle_monatskopf($datum) {
83     switch ($datum[1]) {
84         case 1:
85             $tbl_head = 'Januar_'. $datum[2];
86             break;
87         case 2:
88             $tbl_head = 'Februar_'. $datum[2];
89             break;
90         case 3:
91             $tbl_head = 'M&#228;rz_'. $datum[2];
92             break;
93         case 4:
94             $tbl_head = 'April_'. $datum[2];
95             break;
96         case 5:
97             $tbl_head = 'Mai_'. $datum[2];
98             break;
99         case 6:
100             $tbl_head = 'Juni_'. $datum[2];
101             break;
102         case 7:
103             $tbl_head = 'Juli_'. $datum[2];
104             break;
105         case 8:
106             $tbl_head = 'August_'. $datum[2];
107             break;
108         case 9:
109             $tbl_head = 'September_'. $datum[2];
110             break;
111         case 10:
112             $tbl_head = 'Oktober_'. $datum[2];
113             break;
114         case 11:
115             $tbl_head = 'November_'. $datum[2];
116             break;
117         case 12:
118             $tbl_head = 'Dezember_'. $datum[2];
119             break;
120         case 13:
121             $tbl_head = 'Januar_'. ($datum[2]+1);
122             break;
123     }
124     $this->tblkopfmon = $tbl_head;
125 }
126 }

```

F.3. Quelltext der Geldfunktionen Klasse

Listing F.3: Quelltext der Geldfunktionen Klasse

```
1 class changeto-eur {
2
3   var $preis_eur;
4
5   function eurpreis($waehrung, $altpreis) {
6       /* Wechselkurs holen, Preis ermitteln */
7       $query = "SELECT_umrechnung_FROM_mon_wechselkurs_WHERE
8       waehr.alt=$waehrung'_LIMIT_1";
9       $result = mysql_query($query);
10      while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
11          $wechselk = $line['umrechnung'];
12          $this->preis_eur = round($altpreis*$wechselk, 2);
13      }
14  }
15
16 }
```

F.4. Quelltext der Datensätze Klasse

Listing F.4: Quelltext der Datensätze Klasse

```

1 class datensaetze {
2
3     var $stammdaten_land;
4     var $stammdaten_region;
5     var $stammdaten_ort;
6     var $fh_land;
7     var $fh_region;
8     var $fh_ort;
9     var $desc_haus;
10    var $desc_haus_komplett;
11    var $desc_wohnung;
12    var $desc_wohnung_komplett;
13    var $merkmale_array;
14    var $anzahl_array;
15    var $entfernung_array;
16    var $buchoption_array;
17    var $personenanzahl;
18    var $kinderanzahl;
19    var $hausbilder;
20    var $hausbilder_nummer;
21    var $hausbild_einzelbild;
22    var $objektname;
23    var $objektquali;
24    var $wohnungsbilder;
25    var $wohnungsbilder_nummer;
26    var $wohnungsbilder_einzelbild;
27    var $zubs;
28    var $zubs_incl;
29    var $zubs_vo;
30    var $zubs_zuvo;
31
32    function stammdaten_holen($objreferenzcode, $objkategoriecode) {
33        $stammdaten_holen = "SELECT_*_FROM_obj_stammdaten_WHERE_ObjReferenz_="
34        $objreferenzcode.'_AND_Kategorie_='.$objkategoriecode.'_LIMIT_1';
35        $stammdaten_abfragen = mysql_query($stammdaten_holen);
36        while($line = mysql_fetch_array($stammdaten_abfragen, MYSQL_ASSOC)) {
37            $this->stammdaten_land = $line['Land'];
38            $this->stammdaten_region = $line['Region'];
39            $this->stammdaten_ort = $line['Ort'];
40        }
41    }
42
43    function land_holen($landescode) {
44        /*$pfad = getenv("DOCUMENT_ROOT");
45        $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
46        include $datenbankzugriff; */
47        $land_holen = "SELECT_Land_FROM_ref_land_WHERE_IDLand_=".$landescode."";
48        $land_abfragen = mysql_query($land_holen);
49        while($line = mysql_fetch_array($land_abfragen, MYSQL_ASSOC)) {
50            $this->fh_land = utf8_encode($line['Land']);
51        }
52    }
53
54    function region_holen($regionscode) {
55        $region_holen = "SELECT_Region_FROM_ref_region_WHERE
56        IDRegion_=".$regionscode."_LIMIT_1";
57        $region_abfragen = mysql_query($region_holen);
58        while($line = mysql_fetch_array($region_abfragen, MYSQL_ASSOC)) {
59            $this->fh_region = utf8_encode($line['Region']);
60        }
61    }

```

```

62
63 function ort_holen($sortscore) {
64     $ort_holen = "SELECT _Ort _FROM _ref_ort _WHERE _IDOrt _=_ $sortscore _LIMIT _1";
65     $ort_abfragen = mysql_query($ort_holen);
66     while($line = mysql_fetch_array($ort_abfragen, MYSQL_ASSOC)) {
67         $this->fh_ort = utf8_encode($line['Ort']);
68     }
69 }
70
71 function haustext_holen($objreferenzcode) {
72     $haustext_holen = "SELECT _Beschreibung _FROM _obj_haustext _WHERE
73     _ObjReferenz _=_ '$objreferenzcode' _LIMIT _1";
74     $haustext_abfragen = mysql_query($haustext_holen);
75     if (mysql_num_rows($haustext_abfragen) < 1 ) {
76         $this->desc_haus = 'Zu diesem Eintrag ist keine Hausbeschreibung vorhanden!';
77     }
78     else {
79         while($line = mysql_fetch_array($haustext_abfragen, MYSQL_ASSOC)) {
80             $desc_haus = substr($line['Beschreibung'],0,107);
81             $this->desc_haus_komplett = utf8_encode($line['Beschreibung']);
82             $this->desc_haus = utf8_encode($desc_haus.'...');
83         }
84     }
85 }
86
87 function wohnungstext_holen($objreferenzcode, $objkategoriecode) {
88     $wohnungstext_holen = "SELECT _Beschreibung _FROM _obj_wohnungstext _WHERE
89     _ObjReferenz _=_ '$objreferenzcode' _AND _Kategorie _=_ '$objkategoriecode' _LIMIT _1";
90     $wohnungstext_abfragen = mysql_query($wohnungstext_holen);
91     if (mysql_num_rows($wohnungstext_abfragen) < 1 ) {
92         $this->desc_wohnung =
93         'Zu diesem Eintrag ist keine Wohnungsbeschreibung vorhanden!';
94     }
95     else {
96         while($line = mysql_fetch_array($wohnungstext_abfragen, MYSQL_ASSOC)) {
97             $desc_wohnung = substr($line['Beschreibung'],0,107);
98             $this->desc_wohnung_komplett = utf8_encode($line['Beschreibung']);
99             $this->desc_wohnung = utf8_encode($desc_wohnung.'...');
100         }
101     }
102 }
103
104 function merkmale_holen($objreferenzcode, $objkategoriecode) {
105     $merkmale_holen = "SELECT _* _FROM _obj_merkmale _WHERE
106     _ObjReferenz _=_ '$objreferenzcode' _AND _Kategorie _=_ '$objkategoriecode' _LIMIT _1";
107     $merkmale_abfragen = mysql_query($merkmale_holen);
108     while($line = mysql_fetch_array($merkmale_abfragen, MYSQL_ASSOC)) {
109         $this->merkmale_array[0] = '';
110         if ($line['Anlage'] == 1) {
111             $this->merkmale_array[0] = 'Anlage';
112         }
113         $this->merkmale_array[1] = '';
114         if ($line['Badelandschaft'] == 1) {
115             $this->merkmale_array[1] = 'Badelandschaft';
116         }
117         $this->merkmale_array[2] = '';
118         if ($line['Rollstuhlgeeignet'] == 1) {
119             $this->merkmale_array[2] = 'Rollstuhlgeeignet';
120         }
121         $this->merkmale_array[3] = '';
122         if ($line['Boot'] == 1) {
123             $this->merkmale_array[3] = 'Boot';
124         }
125         $this->merkmale_array[4] = '';
126         if ($line['Caravanpark'] == 1) {
127             $this->merkmale_array[4] = 'Caravanpark';

```

```
128     }
129     $this->merkmale_array[5] = '';
130     if ($line['Endreinigung'] == 1) {
131         $this->merkmale_array[5] = 'Endreinigung';
132     }
133     $this->merkmale_array[6] = '';
134     if ($line['Garage'] == 1) {
135         $this->merkmale_array[6] = 'Garage';
136     }
137     $this->merkmale_array[7] = '';
138     if ($line['Geschirrspuerer'] == 1) {
139         $this->merkmale_array[7] = 'Geschirrsp&#252;ler';
140     }
141     $this->merkmale_array[8] = '';
142     if ($line['Golf'] == 1) {
143         $this->merkmale_array[8] = 'Golf';
144     }
145     $this->merkmale_array[9] = '';
146     if ($line['Haustier'] != 0) {
147         $this->merkmale_array[9] = 'Haustier';
148     }
149     $this->merkmale_array[10] = '';
150     if ($line['Kamin'] == 1) {
151         $this->merkmale_array[10] = 'Kamin';
152     }
153     $this->merkmale_array[11] = '';
154     if ($line['Kinderbetreuung'] == 1) {
155         $this->merkmale_array[11] = 'Kinderbetreuung';
156     }
157     $this->merkmale_array[12] = '';
158     if ($line['Pool'] == 1) {
159         $this->merkmale_array[12] = 'Pool';
160     }
161     $this->merkmale_array[13] = '';
162     if ($line['Sauna'] == 1) {
163         $this->merkmale_array[13] = 'Sauna';
164     }
165     $this->merkmale_array[14] = '';
166     if ($line['Schwimmbad'] == 1) {
167         $this->merkmale_array[14] = 'Schwimmbad';
168     }
169     $this->merkmale_array[15] = '';
170     if ($line['Solarium'] == 1) {
171         $this->merkmale_array[15] = 'Solarium';
172     }
173     $this->merkmale_array[16] = '';
174     if ($line['TV'] == 1) {
175         $this->merkmale_array[16] = 'TV';
176     }
177     $this->merkmale_array[17] = '';
178     if ($line['Waschmaschine'] == 1) {
179         $this->merkmale_array[17] = 'Waschmaschine';
180     }
181     $this->merkmale_array[18] = '';
182     if ($line['Wellness'] == 1) {
183         $this->merkmale_array[18] = 'Wellness';
184     }
185     $this->merkmale_array[19] = '';
186     if ($line['Whirlpool'] == 1) {
187         $this->merkmale_array[19] = 'Whirlpool';
188     }
189
190     $this->anzahl_array[0] = '';
191     if ($line['Personen'] != 0) {
192         if ($line['Personen'] > 1) {
193             $this->anzahl_array[0] = $line['Personen'].'_Personen';
```

```

194     }
195     else {
196         $this->anzahl_array[0] = $line['Personen']. 'Person';
197     }
198 }
199 $this->anzahl_array[1] = '';
200 if ($line['Zimmer'] != 0) {
201     $this->anzahl_array[1] = $line['Zimmer']. 'Zimmer';
202 }
203 $this->anzahl_array[2] = '';
204 if ($line['Schlafraeume'] != 0) {
205     if ($line['Schlafraeume'] > 1) {
206         $this->anzahl_array[2] = $line['Schlafraeume']. 'Schlafr#228;ume';
207     }
208     else {
209         $this->anzahl_array[2] = $line['Schlafraeume']. 'Schlafraum';
210     }
211 }
212
213 $this->entfernung_array[0] = '';
214 if ($line['NaehAngeln'] != 0) {
215     $this->entfernung_array[0] = $line['NaehAngeln']. 'm_zum_Angelplatz';
216 }
217 $this->entfernung_array[1] = '';
218 if ($line['NaehBaden'] != 0) {
219     $this->entfernung_array[1] = $line['NaehBaden']. 'm_zum_Strand';
220 }
221 $this->entfernung_array[2] = '';
222 if ($line['NaehMeer'] != 0) {
223     $this->entfernung_array[2] = $line['NaehMeer']. 'm_bis_zum_Meer';
224 }
225 $this->entfernung_array[3] = '';
226 if ($line['NaehSkigebiet'] != 0) {
227     $this->entfernung_array[3] = $line['NaehSkigebiet']. 'm_zum_Skigebiet';
228 }
229 $this->entfernung_array[4] = '';
230 if ($line['NaehFjord'] != 0) {
231     $this->entfernung_array[4] = $line['NaehFjord']. 'm_zum_n#228;chsten_Fjord';
232 }
233
234 $this->buchoption_array[0] = '';
235 if ($line['FlexBooking'] == 1) {
236     $this->buchoption_array[0] = 'Flexibel_buchbar';
237 }
238 $this->buchoption_array[1] = '';
239 if ($line['ShortBreak'] == 1) {
240     $this->buchoption_array[1] = 'Kurzaufenthalte_m#246;glich';
241 }
242 }
243 }
244
245 function personenanzahl_holen($objreferenzcode, $objkategoriecode) {
246     $personen_holen = "SELECT_Personen,_Kinder_FROM_obj_merkmale_WHERE
247     _ObjReferenz=_ '$objreferenzcode' _AND_Kategorie=_ '$objkategoriecode'
248     _LIMIT_1";
249     $personen_abfragen = mysql_query($personen_holen);
250     if (mysql_num_rows($personen_abfragen) == 0) {
251         $personenanzahl = 2;
252         $kinderanzahl = 0;
253     }
254     else {
255         while($line = mysql_fetch_array($personen_abfragen, MYSQL_ASSOC)) {
256             $personenanzahl = $line['Personen'];
257             $kinderanzahl = $line['Kinder'];
258         }
259     }

```



```

260     $this->personenanzahl = $personenanzahl;
261     $this->kinderanzahl = $kinderanzahl;
262 }
263
264 function hausname_holen($objreferenzcode, $objkategoriecode) {
265     $query = "SELECT_Hausname_FROM_obj_merkmale_WHERE
266     ObjReferenz=$_' $objreferenzcode ' _AND_Kategorie=$_' $objkategoriecode '
267     LIMIT_1";
268     $result = mysql_query($query);
269     if (mysql_num_rows($result) == 0) {
270         $objektnamen = '';
271     }
272     else {
273         while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
274             $objektnamen = utf8_encode($line['Hausname']);
275         }
276         if ($objektnamen == 0) {
277             $objektnamen = '&nbsp;';
278         }
279     }
280     $this->objektnamen = $objektnamen;
281 }
282
283 function quali_holen($objreferenzcode, $objkategoriecode) {
284     $query = "SELECT_CevalA, _CevalB, _CevalC, _CevalD, _CevalE, _CevalF, _Quali_FROM
285     obj_merkmale_WHERE_ObjReferenz=$_' $objreferenzcode ' _AND
286     Kategorie=$_' $objkategoriecode ' _LIMIT_1";
287     $result = mysql_query($query);
288     if (mysql_num_rows($result) == 0) {
289         $quali = array(0,0,0,0,0,0,0);
290     }
291     else {
292         while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
293             $quali = $line;
294         }
295         /* if ($personenanzahl == 0) {
296             $personenanzahl = 3; //Krücke wegen fehlender Wolters Personenanzahl
297         } */
298     }
299     $this->objektquali = $quali;
300 }
301
302 function hausbilder_holen($objreferenzcode) {
303     $hausbilder_holen = "SELECT_*_FROM_obj_hauspic_WHERE
304     ObjReferenz=$_' $objreferenzcode ' _ORDER_BY_Bildnummer_LIMIT_50";
305     $hausbilder_abfragen = mysql_query($hausbilder_holen) or
306     die("Anfrage_gescheitert:" . mysql_error());
307     if (mysql_num_rows($hausbilder_abfragen) < 1 ) {
308         //echo 'Zu diesem Eintrag sind keine Hausbilder vorhanden!<br>';
309     }
310     else {
311         $i_bild = 0;
312         $anzahl_bilder_haus = mysql_num_rows($hausbilder_abfragen);
313         while($line_bilder_haus = mysql_fetch_array($hausbilder_abfragen,
314         MYSQL_ASSOC)) {
315             $hausbilder[$i_bild] = $line_bilder_haus['Bildlink'];
316             $hausbilder_nummer[$i_bild] = $line_bilder_haus['Bildnummer']-1;
317             $i_bild++;
318         }
319         $this->hausbilder = $hausbilder;
320         $this->hausbilder_nummer = $hausbilder_nummer;
321         //Soll das Auftauchen von Sanitärbildern minimieren.
322         //Begrenzung da diese meist höher nummeriert.
323         $ok = false;
324         $zahl = 0;
325         while ($ok != true AND $zahl < 50) {

```

```

326     $imagearray = getimagesize($hausbilder[$zahl]);
327     if (is_array($imagearray)) {
328         $ok = true;
329         $this->hausbild_einzelbild = $hausbilder[$zahl];
330     }
331     $zahl++;
332 }
333     function wohnungsbilder_holen($objreferenzcode, $objkategoriecode) {
334         $wohnungsbilder_holen = "SELECT_Bildlink, _Bildnummer FROM_obj_wohnungspic
335 WHERE_obj_wohnungspic.ObjReferenz=_ '$objreferenzcode' _AND_obj_wohnungspic.
336 Kategorie=_ '$objkategoriecode' _ORDER_BY_Bildnummer_LIMIT_50";
337         $wohnungsbilder_abfragen = mysql_query($wohnungsbilder_holen) or
338         die("Anfrage_gescheitert:" . mysql_error());
339         if (mysql_num_rows($wohnungsbilder_abfragen) < 1 ) {
340             //echo 'Zu diesem Eintrag sind keine Wohnungsbilder vorhanden!<br>';
341         }
342     else {
343         $i_bild = 0;
344         $anzahl_bilder_wohnung = mysql_num_rows($wohnungsbilder_abfragen);
345         while($line_bilder_wohnung = mysql_fetch_array($wohnungsbilder_abfragen ,
346 MYSQLASSOC)) {
347             $wohnungsbilder[$i_bild] = $line_bilder_wohnung['Bildlink'];
348             $wohnungsbilder_nummer[$i_bild] = $line_bilder_wohnung['Bildnummer']-1;
349             $i_bild++;
350         }
351         $this->wohnungsbilder = $wohnungsbilder;
352         $this->wohnungsbilder_nummer = $wohnungsbilder_nummer;
353         //Soll das Auftauchen von Sanitärbildern minimieren.
354         //Begrenzung da diese meist höher nummeriert.
355         if ($anzahl_bilder_wohnung > 3) {
356             $zufallsbild = rand(0, $anzahl_bilder_wohnung-3);
357             $this->wohnungsbild_einzelbild = $wohnungsbilder[$zufallsbild];
358         }
359         else {
360             $zufallsbild = rand(0, $anzahl_bilder_wohnung-1);
361             $this->wohnungsbild_einzelbild = $wohnungsbilder[$zufallsbild];
362         }
363     }
364 }
365
366 function preise_holen () {
367     //Muss hier nicht Referenz- und Kategoriecode übergeben werden?
368     if ($vakanzabfrage_datum != '') {
369         $holen_preise = "SELECT_Preis FROM_obj_preise WHERE
370 WHERE_ObjReferenz=_ '$objkategoriecode' _AND_Kategorie=_ '$objkategoriecode' _AND
371 Aufenthalt=_ '$aufenthaltsdauer' _AND_Datumvon<=_ '$vakanzabfrage_datum'
372 LIMIT_1";
373         $abfragen_preise = mysql_query($holen_preise);
374         while($line_preise = mysql_fetch_array($abfragen_preise, MYSQLASSOC)) {
375             $preisaktuell = $line_preise['Preis'];
376         }
377         echo '<p>ab_'. $vakanzabfrage_datum . '</p>';
378         echo '<p>'. $aufenthaltsdauer . '_Tage</p>';
379         echo '<p>Preis:_'. $preisaktuell . '&#128;</p>';
380     }
381 }
382
383 function zusatzleistungen_holen($anbieter, $objreferenzcode, $objkategoriecode) {
384     /* alle Zusatzleistungen */
385     $zubs_holen = "SELECT_Leistung, _Preis, _Waehrung, _Serviceart, _Berechnung
386 FROM_obj_zusatzleistungen WHERE_ID_Anbieter=_ '$anbieter' _AND
387 ObjReferenz=_ '$objreferenzcode' _AND_Kategorie=_ '$objkategoriecode'
388 LIMIT_50";
389     $zubs_abfragen = mysql_query($zubs_holen);
390     $i = 0;
391     if (mysql_num_rows($zubs_abfragen) < 1) {

```

```

392     $zubs = 'NULL';
393 }
394 else {
395     while($zubs_line = mysql_fetch_array($zubs_abfragen, MYSQL_ASSOC)) {
396         $zubs[$i] = $zubs_line;
397         $i++;
398     }
399 }
400 $this->zubs = $zubs;
401 }
402
403 function zusatzleistungen_holen_getrennt($anbietercode,
404 $objreferenzcode, $objkategoriecode) {
405     /* Im Preis inbegriffen */
406     $zubs_holen_incl = "SELECT_*_FROM_obj_zusatzleistungen WHERE
407     IDAnbieter=$anbietercode AND_ObjReferenz=$objreferenzcode AND
408     Kategorie=$objkategoriecode AND_Serviceart LIKE
409     '%ImPreisInbegriffen%' LIMIT 20";
410     $zubs_abfragen_incl = mysql_query($zubs_holen_incl);
411     $i_db_zubs = 0;
412     if (mysql_num_rows($zubs_abfragen_incl) < 1) {
413         $zubs_incl[$i_db_zubs][0] = '-';
414     }
415     while($zubs_incl_line = mysql_fetch_array($zubs_abfragen_incl,
416     MYSQL_ASSOC)) {
417         $zubs_incl[$i_db_zubs] = array($zubs_incl_line['Leistung'],
418         $zubs_vo_line['Preis'], $zubs_incl_line['Waehrung'],
419         $zubs_incl_line['Serviceart'], $zubs_incl_line['Berechnung']);
420         $i_db_zubs++;
421     }
422
423     /* Vor Ort zu bezahlen */
424     $zubs_holen_vo = "SELECT_*_FROM_obj_zusatzleistungen WHERE
425     IDAnbieter=$anbietercode AND_ObjReferenz=$objreferenzcode AND
426     Kategorie=$objkategoriecode AND_Serviceart LIKE
427     '%VorOrtZuBezahlen%' LIMIT 20";
428     $zubs_abfragen_vo = mysql_query($zubs_holen_vo);
429     $i_db_zubs = 0;
430
431     if (mysql_num_rows($zubs_abfragen_vo) < 1) {
432         $zubs_vo[$i_db_zubs][0] = '-';
433     }
434     while($zubs_vo_line = mysql_fetch_array($zubs_abfragen_vo, MYSQL_ASSOC)) {
435         $zubs_vo[$i_db_zubs] = array($zubs_vo_line['Leistung'],
436         $zubs_vo_line['Preis'], $zubs_vo_line['Waehrung'],
437         $zubs_vo_line['Serviceart'], $zubs_vo_line['Berechnung']);
438         $i_db_zubs++;
439     }
440
441     /* buchbar, zahlbar vor Ort */
442     $zubs_holen_zuvo = "SELECT_*_FROM_obj_zusatzleistungen WHERE
443     IDAnbieter=$anbietercode AND_ObjReferenz=$objreferenzcode AND
444     Kategorie=$objkategoriecode AND (Serviceart LIKE
445     '%buchbar, zahlbar vor Ort%' OR_Serviceart LIKE '%Optional vor Ort%')
446     LIMIT 20";
447     $zubs_abfragen_zuvo = mysql_query($zubs_holen_zuvo);
448     $i_db_zubs = 0;
449
450     if (mysql_num_rows($zubs_abfragen_zuvo) < 1) {
451         $zubs_zuvo[$i_db_zubs][0] = '-';
452     }
453     while($zubs_zuvo_line = mysql_fetch_array($zubs_abfragen_zuvo, MYSQL_ASSOC)) {
454         $zubs_zuvo[$i_db_zubs] = array($zubs_zuvo_line['Leistung'],
455         $zubs_zuvo_line['Preis'], $zubs_zuvo_line['Waehrung'],
456         $zubs_zuvo_line['Serviceart'], $zubs_zuvo_line['Berechnung']);
457         $i_db_zubs++;

```

```
458     }
459     $this->zubs_incl = $zubs_incl;
460     $this->zubs_vo = $zubs_vo;
461     $this->zubs_zuvo = $zubs_zuvo;
462 }
463
464 }
```

F.5. Quelltext der Suchfunktionen Klasse

Listing F.5: Quelltext der Suchfunktionen Klasse

```

1 class suchfunktionen {
2
3   var $mmsqlstring;
4   function merkmaltosqlquery() {
5     /* --- Merkmalsarray aufbauen --- */
6     if ($_SESSION['allgemein'][3][0] == '1') {
7       $mm[9] = 'AND_Haustier=1';
8     }
9     else {
10      $mm[9] = '';
11    }
12    if ($_SESSION['allgemein'][3][1] == '1') {
13      $mm[2] = 'AND_Rollstuhlgeeignet=1';
14    }
15    else {
16      $mm[2] = '';
17    }
18    if ($_SESSION['allgemein'][3][2] == '1') {
19      $mm[18] = 'AND_Waschmaschine=1';
20    }
21    else {
22      $mm[18] = '';
23    }
24    if ($_SESSION['allgemein'][3][3] == '1') {
25      $mm[7] = 'AND_Geschirrspueler=1';
26    }
27    else {
28      $mm[7] = '';
29    }
30    if ($_SESSION['allgemein'][3][4] == '1') {
31      $mm[17] = 'AND_TV=1';
32    }
33    else {
34      $mm[17] = '';
35    }
36    if ($_SESSION['allgemein'][3][5] == '1') {
37      $mm[20] = 'AND_Whirlpool=1';
38    }
39    else {
40      $mm[20] = '';
41    }
42    if ($_SESSION['allgemein'][3][6] == '1') {
43      $mm[8] = 'AND_Golf=1';
44    }
45    else {
46      $mm[8] = '';
47    }
48    if ($_SESSION['allgemein'][3][7] == '1') {
49      $mm[10] = 'AND_Kamin=1';
50    }
51    else {
52      $mm[10] = '';
53    }
54    if ($_SESSION['allgemein'][3][8] != 'ka' AND $_SESSION['allgemein'][3][8] >= 10)
55    {
56      $mm[21] = 'AND_NaeheAngeln>0AND_NaeheAngeln<=.';
57      $_SESSION['allgemein'][3][8];
58    }
59    else {
60      $mm[21] = '';
61    }

```

```

62     if ($SESSION['allgemein'][3][9] != 'kA' AND $SESSION['allgemein'][3][9] >= 10)
63     {
64         $mm[22] = 'AND_NaheBaden>0AND_NaheBaden<=0';
65         $SESSION['allgemein'][3][9];
66     }
67     else {
68         $mm[22] = '';
69     }
70     if ($SESSION['allgemein'][3][10] != 'kA' AND $SESSION['allgemein'][3][10] >= 10)
71     {
72         $mm[25] = 'AND_NaheMeer>0AND_NaheMeer<=0';
73         $SESSION['allgemein'][3][10];
74     }
75     else {
76         $mm[25] = '';
77     }
78     if ($SESSION['allgemein'][3][11] != 'kA' AND $SESSION['allgemein'][3][11] >= 10)
79     {
80         $mm[24] = 'AND_NaheSkigebiet>0AND_NaheSkigebiet<=0';
81         $SESSION['allgemein'][3][11];
82     }
83     else {
84         $mm[24] = '';
85     }
86     if ($SESSION['allgemein'][4][0] == '1') {
87         $mm[13] = 'AND_Sauna=1';
88     }
89     else {
90         $mm[13] = '';
91     }
92     if ($SESSION['allgemein'][4][1] == '1') {
93         $mm[16] = 'AND_Solarium=1';
94     }
95     else {
96         $mm[16] = '';
97     }
98     if ($SESSION['allgemein'][4][2] == '1') {
99         $mm[19] = 'AND_Wellness=1';
100    }
101    else {
102        $mm[19] = '';
103    }
104    if ($SESSION['allgemein'][4][3] == '1') {
105        $mm[31] = 'AND_Whirlpool=1';
106    }
107    else {
108        $mm[31] = '';
109    }
110    if ($SESSION['allgemein'][4][4] == '1') {
111        $mm[14] = 'AND_Schwimmbad=1';
112    }
113    else {
114        $mm[14] = '';
115    }
116    if ($SESSION['allgemein'][4][5] == '1') {
117        $mm[1] = 'AND_Badelandschaft=1';
118    }
119    else {
120        $mm[1] = '';
121    }
122    if ($SESSION['allgemein'][4][6] == '1') {
123        $mm[3] = 'AND_Boot=1';
124    }
125    else {
126        $mm[3] = '';
127    }

```

```
128     if ($_SESSION['allgemein'][4][7] == '1') {
129         $mm[6] = 'AND_Garage=1';
130     }
131     else {
132         $mm[6] = '';
133     }
134     if ($_SESSION['allgemein'][4][8] == '1') {
135         $mm[11] = 'AND_Kinderbetreuung=1';
136     } else {
137         $mm[11] = '';
138     }
139     if (isset($_SESSION['allgemein'][2][0]) AND $_SESSION['allgemein'][2][0] != 'kA'
140     AND $_SESSION['allgemein'][2][0] != '') {
141         $mm[28] = 'AND_Personen=>'.$_SESSION['allgemein'][2][0];
142     }
143     else {
144         $mm[28] = '';
145     }
146     if (isset($_SESSION['allgemein'][2][1]) AND $_SESSION['allgemein'][2][1] != 'kA'
147     AND $_SESSION['allgemein'][2][1] != '') {
148         $mm[29] = 'AND_Zimmer=>'.$_SESSION['allgemein'][2][1];
149     }
150     else {
151         $mm[29] = '';
152     }
153     if (isset($_SESSION['allgemein'][2][2]) AND $_SESSION['allgemein'][2][2] != 'kA'
154     AND $_SESSION['allgemein'][2][2] != '') {
155         $mm[30] = 'AND_Schlafräume=>'.$_SESSION['allgemein'][2][2];
156     }
157     else {
158         $mm[30] = '';
159     }
160
161     $this->mmsqlstring = implode(" ", $mm);
162 }
163
164 }
```

G. Quelltexte der Buchungsmaske

G.1. Suchmaske

Listing G.1: Quelltext der Suchmaske

```

1 <?php
2 $sessionid = $_REQUEST['sid'];
3 if ($sessionid != '' AND strlen($sessionid) == 32) {
4     session_id($sessionid);
5 }
6 session_start();
7 /* Initialisierung SESSION Arrays */
8 $_SESSION['navigation'] = array('kA', 'kA');
9 $_SESSION['zusatz'] = '';
10 if ($_GET['s'] == '1') {
11     $suchtyp = 'profi';
12 }
13 else {
14     $suchtyp = 'standard';
15 }
16 $fehler = $_SESSION['meldung'];
17 ?>
18 <!DOCTYPE html PUBLIC "-//W3C//DTD_XHTML1.0_Transitional//EN" "http://www.w3.org/TR/
19 xhtml1/DTD/xhtml1-transitional.dtd">
20 <html>
21     <html xmlns="http://www.w3.org/1999/xhtml">
22     <head>
23         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
24         <noscript>
25             <!-- Ihr Browser unterst#252;tzt kein Javascript. Leite auf Non Javascript
26              Version um<br> /-->
27             <meta http-equiv="refresh" content="0; URL=index-nj.php">
28         </noscript>
29         <title>Abfragetests</title>
30         <link href="../css/ferienhaus_ibe.css" rel="stylesheet" type="text/css">
31         <script type="text/javascript">
32             function XmlHttp( ){
33                 this.CreateXmlHttpRequest = CreateXmlHttpRequest;
34                 this.GetUrlContent           = GetUrlContent;
35                 this.GetResponseText        = GetResponseText;
36                 this.GetReadyState          = GetReadyState;
37                 this.HttpMethod = 'GET'; // default
38                 this.objXmlHttp = this.CreateXmlHttpRequest();
39             }
40
41             // Initialize XMLHttpRequest
42             function CreateXmlHttpRequest(){
43                 var xmlhttp=false;
44                 try {
45                     xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
46                 } catch (e) {
47                     try {
48                         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
49                     } catch (E) {
50                         xmlhttp = false;

```



```

51     }
52 }
53
54
55 if (!xmlhttp && typeof XMLHttpRequest!= 'undefined') {
56     xmlhttp = new XMLHttpRequest();
57 }
58     return xmlhttp;
59 }
60
61 var objXMLHttp = new XmlHttp();
62 var objXMLHttp2 = new XmlHttp();
63 var objXMLHttp3 = new XmlHttp();
64 var objXMLHttp4 = new XmlHttp();
65 var objXMLHttp5 = new XmlHttp();
66
67 function GetReadyState( ){
68     return this.objXmlHttp.readyState;
69 }
70
71 function GetResponseText( ){
72     return this.objXmlHttp.responseText;
73 }
74
75
76 // Function performs Get request to absolute url(strUrl)
77 // using XmlHttp object (asynchroni)
78 // Response returned into objResult element using innerHTML.
79 // When state of XmlHttp object is changed – objOnReadyStateChangeFunction called
80 function GetUrlContent( strUrl , objOnReadyStateChangeFunction ){
81     this.objXmlHttp.open(this.HttpMethod, strUrl, true);
82     this.objXmlHttp.setRequestHeader( 'Content-Type', 'text/xml; charset=UTF-8' );
83     if(objOnReadyStateChangeFunction){
84         this.objXmlHttp.onreadystatechange=function(){
85             objOnReadyStateChangeFunction();
86         }
87     }
88     this.objXmlHttp.send( null)
89 }
90
91 //This function is called when we get the data back from the server.
92 function GetResponse( ){
93     if (objXMLHttp.GetReadyState()==4) {
94         // save response in inner html of result object
95         var objMyDiv = document.getElementById( 'MyDiv' );
96         objMyDiv.innerHTML = objXMLHttp.GetResponseText( );
97     }
98 }
99
100 function GetResponse2( ){
101     if (objXMLHttp2.GetReadyState()==4) {
102         // save response in inner html of result object
103         var objMyDiv2 = document.getElementById( 'MyDiv2' );
104         objMyDiv2.innerHTML = objXMLHttp2.GetResponseText( );
105     }
106 }
107
108 function GetResponse3( ){
109     if (objXMLHttp3.GetReadyState()==4) {
110         // save response in inner html of result object
111         var objMyDiv3 = document.getElementById( 'MyDiv3' );
112         objMyDiv3.innerHTML = objXMLHttp3.GetResponseText( );
113     }
114 }
115
116 //Neu 7.2.08

```

```

117     function GetResponse4( ){
118         if (objXMLHttp4.GetReadyState()==4) {
119             // save response in inner html of result object
120             var objMyDiv4 = document.getElementById( 'MyDiv4' );
121             objMyDiv4.innerHTML = objXMLHttp4.GetResponseText( );
122         }
123     }
124
125     //Neu 7.7.08
126     function GetResponse5( ){
127         if (objXMLHttp5.GetReadyState()==4) {
128             // save response in inner html of result object
129             var objMyDiv5 = document.getElementById( 'MyDiv5' );
130             objMyDiv5.innerHTML = objXMLHttp5.GetResponseText( );
131         }
132     }
133
134     function UpdateDiv( FieldValue ){
135         if( FieldValue != '' ){
136             var objMyDiv = document.getElementById( 'MyDiv' );
137             var objMyDiv2 = document.getElementById( 'MyDiv2' );
138             var objMyDiv3 = document.getElementById( 'MyDiv3' );
139             var objMyDiv4 = document.getElementById( 'MyDiv4' );
140             objMyDiv.innerHTML = '<table><tr><td class="geotd1">Region:</td><td>
141 .....<select class="geoselect" name="L2" size="1" style="width:230px">
142 .....<option value="kA">Bitte_Warten</option></select></td></tr></table>';
143             objXMLHttp.GetUrlContent( 'http://' + location.host +
144             '/ferienhaus-suche/ajax/zielsuche.php?L1=' + FieldValue + '&L2=NULL',
145             GetResponse );
146         }
147         return;
148     }
149     function UpdateDiv2( FieldValue2, FieldValue1){
150         if( FieldValue2 != '' && FieldValue1 != '' ){
151             var objMyDiv2 = document.getElementById( 'MyDiv2' );
152             var objMyDiv3 = document.getElementById( 'MyDiv3' );
153             var objMyDiv4 = document.getElementById( 'MyDiv4' );
154             objMyDiv2.innerHTML = '<table><tr><td class="geotd1">Ort:</td><td>
155 .....<select class="geoselect" name="L3" size="1" style="width:230px">
156 .....<option value="kA">Bitte_Warten</option></select></td></tr></table>';
157             objXMLHttp2.GetUrlContent( 'http://' + location.host +
158             '/ferienhaus-suche/ajax/zielsuche.php?L3=NULL&L2=' + FieldValue2 + '&L1=' +
159             FieldValue1 + '&Level=2', GetResponse2 );
160         }
161         return;
162     }
163     function UpdateDiv3( FieldValue3, FieldValue2, FieldValue1){
164         if( FieldValue3 != '' && FieldValue2 != '' && FieldValue1 != '' ){
165             var objMyDiv3 = document.getElementById( 'MyDiv3' );
166             var objMyDiv4 = document.getElementById( 'MyDiv4' );
167             objMyDiv3.innerHTML = 'Bitte_warten';
168             objXMLHttp3.GetUrlContent( 'http://' + location.host +
169             '/ferienhaus-suche/ajax/zielsuche.php?L3=' + FieldValue3 + '&L2=' +
170             FieldValue2 + '&L1=' + FieldValue1 + '&Level=3', GetResponse3 );
171         }
172         return;
173     }
174
175     var objTimer = null;
176     function liveSearchStart( suchchar ) {
177         if (objTimer) {
178             window.clearTimeout( objTimer );
179         }
180         objTimer = window.setTimeout( "UpdateDiv3( document.MyForm.wunschziel.value ,
181 ..... 'SUCHE' , 'NULL' )" , 500 );
182     }

```

```

183
184     function Clear1st() {
185         var objsuchfenster = document.getElementById( 'liveziel' );
186         objsuchfenster.innerHTML = '<input_class="wunschztext" _type="text"
187     .....name="wunschziel" _value="" _maxlength="60"
188     .....onkeypress="liveSearchStart( this.value);"></input>';
189     }
190
191     function Clear(selValue) {
192         var objsuchfenster = document.getElementById( 'liveziel' );
193         var ergmenu = document.getElementById( 'MyDiv3' );
194         var worte = selValue.split ( '-' );
195         objsuchfenster.innerHTML = '<input_class="wunschztext" _type="text"
196     .....name="wunschziel" _value="'+worte[1]+'"' _maxlength="60"
197     .....onkeypress="liveSearchStart( this.value);"></input><input_name="direktsuche"
198     .....type="hidden" _value="'+worte[0]+'"'></input>';
199         ergmenu.innerHTML = '&nbsp';
200     }
201
202     function CheckInput() {
203         var fall1 = true;
204         var fall2 = true;
205         var fall3 = true;
206         var LandMenu = document.getElementById ( 'L1' ).value;
207         var WunschZiel = document.getElementById ( 'wunschziel' ).value;
208         var ObjektDirekt = document.getElementById ( 'objektcode' ).value;
209
210         if (LandMenu == 'kA' || LandMenu == '') {
211             var fall1 = false;
212         }
213         if (WunschZiel == '' || WunschZiel == 'Wunschziel') {
214             var fall2 = false;
215         }
216         if (ObjektDirekt == '') {
217             var fall3 = false;
218         }
219
220         var content = 'Es_wird_nach_Angeboten_gesucht...';
221         document.getElementById ( 'InfoFeld' ).firstChild.data = content;
222         if (!fall1 && !fall2 && !fall3) {
223             var content = 'Bitte_treffen_sie_mindestens_eine_Auswahl!';
224             document.getElementById ( 'InfoFeld' ).firstChild.data = content;
225             return (false);
226         }
227     }
228     </script>
229     <script language="javascript" type="text/javascript" src="cal.js"></script>
230 </head>
231 <body>
232     <?php
233     $i = 0;
234     $pfad = getenv("DOCUMENTROOT");
235     $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
236     include $datenbankzugriff;
237     echo $suchtyp;
238     ?>
239     Kalenderscript ist testweise von http://www.goweb.de!<br>
240     <!-- ***Linkleiste oben*** //-->
241     <div>
242         <?php
243         $seite = 'index';
244         include $pfad.'/include/suche_headlinks.php';
245         ?>
246     </div>
247     <!-- ***Start Haupt Div Container*** //-->
248     <div name="main" id="main">

```

```

249 <form action="uebersicht_ferienhaeuser.php" name="MyForm" id="MyForm"
250 method="post" onsubmit="return _CheckInput();">
251 <div name="headline1" class="headline">
252     Suche
253 </div>
254 <!-- ***Start Auswahlcontainer Land, Region, Ort*** -->
255 <div name="geobox" class="geobox">
256     <!-- ***Landesauswahl*** -->
257     <div name="MyDiv4" id="MyDiv4">
258         <table>
259             <tr> <!-- Land auswählen -->
260                 <td class="geotd1">Land:</td>
261                 <td>
262                     <select name="L1" id="L1" size="1" class="geoselect"
263                         onchange="UpdateDiv(this.value);"> <!-- Auswählen eines der
264                         Menüeinträge führt zu Inhaltsänderung des DIV containers
265                         "MyDiv" -->
266                         <option value="kA">Land w&#228;hlen</option>
267                 <?php
268                     //SQL Abfrage
269                     echo $pfad.'<br>';
270                     echo $datenbankzugriff;
271                     $query = "SELECT _IDLand, _Land FROM _menu_land _$countrycode _ORDER
272     ..... _BY _Land";
273                     $result = mysql_query($query) or die("Anfrage_gescheitert:" .
274                     mysql_error());
275                     //Ergebnis
276                     while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
277                         $landescode = $line["IDLand"];
278                         $landesname = utf8_encode($line["Land"]);
279                         $query_land_objekte = "SELECT _Land FROM _obj_stammdaten _WHERE
280     ..... _Land=_ '$landescode'";
281                         $result_land_objekte = mysql_query($query_land_objekte) or
282                         die("Anfrage_gescheitert:" . mysql_error());
283                         $objekte_land_anzahl = mysql_num_rows($result_land_objekte);
284                         if ($objekte_land_anzahl > 0) {
285                             echo '<option value="' . $line["IDLand"] .
286                             "'>' . utf8_encode($line["Land"] .
287                             ' (' . $objekte_land_anzahl . ')</option>';
288                         }
289                     }
290                     mysql_free_result($result);
291                 ?>
292             </select>
293         </td>
294     </tr>
295 </table>
296 </div>
297 <!-- ***Regionsauswahl*** -->
298 <div name="MyDiv" id="MyDiv">
299     <table>
300         <tr>
301             <td class="geotd1">Region:</td>
302             <td>
303                 <select name="L2" size="1" class="geoselect"
304                     onchange="UpdateDiv2(this.value);">
305                 <option value="kA">Bitte zuerst ein Land ausw&#228;hlen</option>
306                 </select>
307             </td>
308         </tr>
309     </table>
310 </div>
311 <!-- ***Ortsauswahl*** -->
312 <div name="MyDiv2" id="MyDiv2">
313     <table>
314         <tr>

```

```

315         <td class="geotd1">Ort:</td>
316         <td>
317             <select name="L3" size="1" class="geoselect">
318                 <option value="kA">Bitte zuerst ein Land ausw&#228;hlen</option>
319             </select>
320         </td>
321     </tr>
322 </table>
323 </div>
324 <!-- ***Ende Auswahlcontainer Land, Region, Ort*** -->
325 </div>
326
327 <!-- ***Start Reisedatum, Aufenthaltsdauer*** -->
328 <div class="aufenth">
329     <table>
330         <tr>
331             <td class="aufenthtd1">Von:</td>
332             <td>
333                 <input type="text" id="anreise" size="20" maxlength="32" name="L4" />
334                 <a href="JavaScript:Kalender.anzeige(null,null,
335 ..... 'anreise',7,365,'%d.%m.%y')"></a>
336             </td>
337         </tr>
338         <tr>
339             <td class="aufenthtd1">Dauer</td>
340             <td>
341                 <select name="L5" size="1">
342                     <option value="7"> 7</option>
343                     <option value="14">14</option>
344                     <option value="21">21</option>
345                     <option value="28">28</option>
346                 </select> Tage
347             </td>
348         </tr>
349     </table>
350     <input type="checkbox" name="fhprivat" value="privat">Nur Ferienhaus
351     Angebote von Privat<br>
352     <span class="red" id="InfoFeld"><?php echo $fehler[0];?>&nbsp;</span>
353 <!-- ***Ende Reisedatum, Aufenthaltsdauer*** -->
354 </div>
355
356 <div class="headline">
357     Direkt nach einem Zielort oder Objekt suchen
358 </div>
359 <!-- ***Start Direktsuche*** -->
360 <div name="suchfenster" id="suchfenster" class="wunschzbox">
361     <table>
362         <tr>
363             <td class="wunschztd1">
364                 Wunschziel:
365             </td>
366             <td id="liveziel">
367                 <input class="wunschztext" type="text" name="direktsuche"
368                     id="wunschziel" value="Wunschziel" maxlength="60"
369                     onkeypress="liveSearchStart(this.value);
370 ..... " onclick="document.getElementById('wunschziel').value='';"></input>
371                 <input name="direktsuche" type="hidden" value=""></input>
372             </td>
373         </tr>
374     </table>
375 </div>
376 <!-- ***Start Objekt Direktsuche*** -->
377 <div class="objsuchbox">
378     <table>
379         <tr>
380             <td>

```

```

381         Objektcode :
382     </td>
383 </td>
384     <input class="textfeld" type="text" name="objektcode"
385         id="objektcode" maxlength="30" size="15"></input>
386 </td>
387 </tr>
388 </table>
389 </div>
390 <!-- ***Ende Objekt Direktsuche*** /-->
391 <div NAME="MyDiv3" ID="MyDiv3" class="wunscherg1">
392 &nbsp;
393 </div>
394 <!-- **Ende Direktsuche*** -->
395
396 <?php
397 if ( $suchtyp == 'profi' ) {
398 ?>
399     <div class="headline">
400         R&#228;umlichkeiten
401     </div>
402
403     <!-- ***Start Auswahlmenüs Personen , Zimmeranzahl*** /-->
404     <div class="raumbox">
405         <table>
406             <tr>
407                 <td style="width:50px;">Personen:</td>
408                 <td style="width:80px;">
409                     <select class="textfeld" name="L6" size="1" style="width:70px">
410                         <option value="kA" selected="selected">Egal</option>
411                         <option value="1">1</option>
412                         <option value="2">2</option>
413                         <option value="3">3</option>
414                         <option value="4">4</option>
415                         <option value="5">5</option>
416                         <option value="6">6</option>
417                         <option value="7">7</option>
418                         <option value="8">8</option>
419                         <option value="9">9</option>
420                         <option value="10">10</option>
421                         <option value="11">11</option>
422                         <option value="12">12</option>
423                         <option value="13">13</option>
424                         <option value="14">14</option>
425                         <option value="15">15</option>
426                     </select>
427                 </td>
428                 <td style="width:50px;">Zimmer:</td>
429                 <td style="width:80px;">
430                     <select class="textfeld" name="L7" size="1" style="width:70px">
431                         <option value="kA" selected="selected">Egal</option>
432                         <option value="1">1</option>
433                         <option value="2">2</option>
434                         <option value="3">3</option>
435                         <option value="4">4</option>
436                         <option value="5">5</option>
437                         <option value="6">6</option>
438                         <option value="7">7</option>
439                         <option value="8">8</option>
440                         <option value="9">9</option>
441                         <option value="10">10</option>
442                         <option value="11">11</option>
443                         <option value="12">12</option>
444                         <option value="13">13</option>
445                         <option value="14">14</option>
446                         <option value="15">15</option>

```

```

447         </select>
448     </td>
449     <td style="width:50px;">Schlafr&#228;ume:</td>
450     <td style="width:80px;">
451         <select class="textfeld" name="L8" size="1" style="width:70px">
452             <option value="kA" selected="selected">Egal</option>
453             <option value="1">1</option>
454             <option value="2">2</option>
455             <option value="3">3</option>
456             <option value="4">4</option>
457             <option value="5">5</option>
458             <option value="6">6</option>
459             <option value="7">7</option>
460             <option value="8">8</option>
461             <option value="9">9</option>
462             <option value="10">10</option>
463             <option value="11">11</option>
464             <option value="12">12</option>
465             <option value="13">13</option>
466             <option value="14">14</option>
467             <option value="15">15</option>
468         </select>
469     </td>
470 </tr>
471 </table>
472 </div>
473 <!-- ***Ende Auswahlmenüs Personen , Zimmeranzahl*** //-->
474
475 <div class="headline">
476     Ausstattung
477 </div>
478 <!-- 12 Merkmale Kategorie ERWEITERT -->
479 <!-- ***Start Auswahlbereich Ausstattung1*** //-->
480 <div class="ausstattbox1">
481     <table>
482     <tr>
483         <td>Haustier:</td>
484         <td><input type="checkbox" maxlength="20" name="M1"
485             value="1"></input></td>
486         <td>Rollstuhlgeeignet:</td>
487         <td><input type="checkbox" maxlength="20" name="M2"
488             value="1"></input></td>
489     </tr>
490     <tr>
491         <td>Waschmaschine:</td>
492         <td><input type="checkbox" maxlength="20" name="M3"
493             value="1"></input></td>
494         <td>Geschirrsp&#252;ler:</td>
495         <td><input type="checkbox" maxlength="20" name="M4"
496             value="1"></input></td>
497     </tr>
498     <tr>
499         <td>TV:</td>
500         <td><input type="checkbox" maxlength="20" name="M5"
501             value="1"></input></td>
502         <td>Pool:</td>
503         <td><input type="checkbox" maxlength="20" name="M6"
504             value="1"></input></td>
505     </tr>
506     <tr>
507         <td>Golfen:</td>
508         <td><input type="checkbox" maxlength="20" name="M7"
509             value="1"></input></td>
510         <td>Kamin:</td>
511         <td><input type="checkbox" maxlength="20" name="M8"
512             value="1"></input></td>

```

```

513         </tr>
514     </table>
515     <!-- ***Ende Auswahlbereich Ausstattung1*** //-->
516 </div>
517
518 <!-- ***Start Auswahlbereich Ausstattung2*** //-->
519 <div class="ausstattbox2">
520     <table>
521         <tr>
522             <td style="width:50px;">Angeln:</td>
523             <td style="width:50px;">Badem<del>#246</del>lichkeit:</td>
524         </tr>
525         <tr>
526             <td>
527                 <select class="textfield" name="M9" size="1" style="width:100px">
528                     <option value="kA" selected="selected" >Egal</option>
529                     <option value="100">bis 100m</option>
530                     <option value="1000">bis 1000m</option>
531                     <option value="10000">bis 10km</option>
532                 </select>
533             </td>
534             <td>
535                 <select class="textfield" name="M10" size="1" style="width:100px">
536                     <option value="kA" selected="selected" >Egal</option>
537                     <option value="100">bis 100m</option>
538                     <option value="1000">bis 1000m</option>
539                     <option value="10000">bis 10km</option>
540                 </select>
541             </td>
542         </tr>
543         <tr>
544             <td style="width:50px;">Meer:</td>
545             <td style="width:50px;">Skilifte</td>
546         </tr>
547         <tr>
548             <td>
549                 <select class="textfield" name="M11" size="1" style="width:100px">
550                     <option value="kA" selected="selected" >Egal</option>
551                     <option value="100">bis 100m</option>
552                     <option value="1000">bis 1000m</option>
553                     <option value="10000">bis 10km</option>
554                 </select>
555             </td>
556             <td>
557                 <select class="textfield" name="M12" size="1" style="width:100px">
558                     <option value="kA" selected="selected" >Egal</option>
559                     <option value="100">bis 100m</option>
560                     <option value="1000">bis 1000m</option>
561                     <option value="10000">bis 10km</option>
562                 </select>
563             </td>
564         </tr>
565     </table>
566     <!-- ***Ende Auswahlbereich Ausstattung2*** //-->
567 </div>
568
569 <div class="headline">
570     Extras
571 </div>
572
573 <!-- ***Start Auswahlbereich Extras*** //-->
574 <div class="extrasbox">
575     <table style="width:100%;">
576         <tr>
577             <td>Sauna:</td>
578             <td><input type="checkbox" maxlength="20" name="E1">

```



```

579         value="1"></input></td>
580     <td>Solarium:</td>
581     <td><input type="checkbox" maxlength="20" name="E2"
582         value="1"></input></td>
583     <td>Wellness:</td>
584     <td><input type="checkbox" maxlength="20" name="E3"
585         value="1"></input></td>
586 </tr>
587 <tr>
588     <td>Whirlpool:</td>
589     <td><input type="checkbox" maxlength="20" name="E4"
590         value="1"></input></td>
591     <td>Schwimmbad:</td>
592     <td><input type="checkbox" maxlength="20" name="E5"
593         value="1"></input></td>
594     <td>Badelandschaft:</td>
595     <td><input type="checkbox" maxlength="20" name="E6"
596         value="1"></input></td>
597 </tr>
598 <tr>
599     <td>Boot:</td>
600     <td><input type="checkbox" maxlength="20" name="E7"
601         value="1"></input></td>
602     <td>Garage:</td>
603     <td><input type="checkbox" maxlength="20" name="E8"
604         value="1"></input></td>
605     <td>Kinderbetreuung:</td>
606     <td><input type="checkbox" maxlength="20" name="E9"
607         value="1"></input></td>
608 </tr>
609 </table>
610 <!-- ***Ende Auswahlbereich Extras*** /-->
611 </div>
612 <?php
613 }
614 $_SESSION['meldung'][0] = '';
615 ?>
616 <div class="suchbtbox">
617     <input name="eintrakt" type="hidden" value="0"></input>
618     <input name="suche" type="hidden" value="1"></input>
619     <input name="seitenstart" type="hidden" value="1"></input>
620     <input name="sid" type="hidden" value="<?php echo $_session_id();?>"></input>
621     <input type="submit" value="suchen"></input>
622 </div>
623 </form>
624 <!-- ***Ende Haupt Div Container*** /-->
625 </div>
626 </body>
627 </html>

```

G.2. Ergebnisübersicht

Listing G.2: Quelltext der Objektübersicht

```

1 <?php
2 $sessionid = $_REQUEST['sid'];
3 if (!isset($_COOKIE['PHPSESSID'])) {
4     session_id($sessionid);
5 }
6 session_start();
7 $downward = "";
8 $Pfad = getenv("REQUEST_URI");
9 $Pfadteile = explode("/", $Pfad);
10 for ($i=count($Pfadteile)-1;$i>=1;$i--)

```

```

11 {
12     $downward = $downward . " ../";
13 }
14 $pfad = getenv("DOCUMENT_ROOT");
15 $datenbankzugriff = $pfad . '/include/opendb_ferienhaus.php';
16 include $datenbankzugriff;
17 $inc_objektklasse = $pfad . '/include/class_objekte_db.php';
18 include $inc_objektklasse;
19 $inc_suchklasse = $pfad . '/include/class_suche_db.php';
20 include $inc_suchklasse;
21 $tabelle = 'usr_web31_1';
22
23 $suchsession = new suchfunktionen;
24 $querymaxentries = false;
25 require_once("varcheck.php");
26
27 /* Erforderliche Suchparameter */
28 $country = $_SESSION['allgemein'][0][0];
29 $region   = $_SESSION['allgemein'][0][1];
30 $city     = $_SESSION['allgemein'][0][2];
31 $privatsuche = $_SESSION['allgemein'][5][0];
32
33 /* Datums- und Personendaten */
34 $vakanzabfrage_datum = $_SESSION['allgemein'][1][0];
35 $aufenthaltsdauer = $_SESSION['allgemein'][1][1];
36 $anz_personen = $_SESSION['allgemein'][2][0];
37 $anz_zimmer = $_SESSION['allgemein'][2][1];
38 $anz_schlafräume = $_SESSION['allgemein'][2][2];
39
40 /* Ausstattung, Bereich Entfernungsangaben */
41 $entf_angeln = $_SESSION['allgemein'][3][8];
42 $entf_bademoegl = $_SESSION['allgemein'][3][9];
43 $entf_meer = $_SESSION['allgemein'][3][10];
44 $entf_skilifte = $_SESSION['allgemein'][3][11];
45
46 /* Parameter für die Seitenabfrage festlegen */
47 $entries = $_SESSION['navigation'][0];
48 $seitenstart = $_SESSION['navigation'][1];
49 if (isset($_SESSION['navigation'][2]) AND $_SESSION['navigation'][2] != '') {
50     $eintraege = $_SESSION['navigation'][2];
51 }
52 else {
53     $eintraege = 10;
54 }
55
56 /* Auswertung Suchparamter, sql query vorbereiten für Aufenthaltsregion */
57 if ($region == 'ka' OR $region == '' OR $region == '0') {
58     $regionssuche = '_';
59 }
60 else {
61     $regionssuche = '_AND_Region_'. $region;
62 }
63
64 if ($city == 'ka' OR $city == '' OR $city == '0') {
65     $citysuche = '_';
66 }
67 else {
68     $citysuche = '_AND_Ort_'. $city;
69 }
70
71 /* aufbau Merkmalsstring für sql Abfrage */
72 $suchsession->merkmaltostring();
73 $mmsql = $suchsession->mmsqlstring;
74 $mmlaenge = strlen($mmsql);
75
76 /* — Suchtyp festlegen — */

```

```

77 $merkmale_aktiv = 0;
78 if ($mm_laenge > 30) {
79     $merkmale_aktiv = 1;
80 }
81
82 /* **Differenzen in der Anzahl der Objekte wenn mit der Merkmalstabelle referenziert.
83   Scheinbar haben einige Objekte keine Merkmalseinträge** */
84 /* **Aufspaltung in Suche mit und ohne Merkmale, Privatangebote, Alles** */
85 if ($privatsuche == 'privat') {
86     if ($merkmale_aktiv == 1) {
87         echo 'Merkmale<br>';
88         $query = "SELECT_*_FROM_$tabelle.obj_stammdaten,_$tabelle.obj_merkmale_WHERE
89         ObjReferenz_LIKE_'%ONFH%'
90         AND_obj_merkmale.ObjReferenz=obj_stammdaten.ObjReferenz_AND
91         obj_merkmale.Kategorie=obj_stammdaten.Kategorie
92         AND_Land=_'$country'_$regionssuche_$citysuche_$mmsql_ORDER_BY
93         $tabelle.obj_stammdaten.ObjReferenz_LIMIT_$entries,$eintraege";
94         $query_all = "SELECT_*_FROM_$tabelle.obj_stammdaten,_$tabelle.obj_merkmale_WHERE
95         ObjReferenz_LIKE_'%ONFH%'_AND
96         obj_merkmale.ObjReferenz=obj_stammdaten.ObjReferenz_AND
97         obj_merkmale.Kategorie=obj_stammdaten.Kategorie
98         AND_Land=_'$country'_$regionssuche_$citysuche_$mmsql_ORDER_BY
99         $tabelle.obj_stammdaten.ObjReferenz_LIMIT_500";
100         echo $query;
101         print_r($_SESSION['allgemein']);
102     }
103     else {
104         //echo 'Keine Merkmale<br>';
105         $query = "SELECT_*_FROM_$tabelle.obj_stammdaten_WHERE_Land=_'$country'
106         $regionssuche_$citysuche_AND
107         ObjReferenz_LIKE_'%ONFH%'_ORDER_BY
108         $tabelle.obj_stammdaten.ObjReferenz_LIMIT_$entries,$eintraege";
109         $query_all = "SELECT_*_FROM_$tabelle.obj_stammdaten_WHERE_Land=_'$country'
110         $regionssuche_$citysuche_AND
111         ObjReferenz_LIKE_'%ONFH%'_ORDER_BY_$tabelle.obj_stammdaten.ObjReferenz_LIMIT_500";
112     }
113 }
114 else {
115     if ($merkmale_aktiv == 1) {
116         echo 'Merkmale<br>';
117         $query = "SELECT_*_FROM_$tabelle.obj_stammdaten,_$tabelle.obj_merkmale_WHERE
118         obj_merkmale.ObjReferenz=obj_stammdaten.ObjReferenz_AND
119         obj_merkmale.Kategorie=obj_stammdaten.Kategorie
120         AND_Land=_'$country'_$regionssuche_$citysuche_$mmsql_ORDER_BY
121         $tabelle.obj_stammdaten.ObjReferenz_LIMIT_$entries,$eintraege";
122         $query_all = "SELECT_*_FROM_$tabelle.obj_stammdaten,_$tabelle.obj_merkmale_WHERE
123         obj_merkmale.ObjReferenz=obj_stammdaten.ObjReferenz_AND
124         obj_merkmale.Kategorie=obj_stammdaten.Kategorie
125         AND_Land=_'$country'_$regionssuche_$citysuche_$mmsql_ORDER_BY
126         $tabelle.obj_stammdaten.ObjReferenz_LIMIT_500";
127         echo $query;
128         print_r($_SESSION['allgemein']);
129     }
130     else {
131         //echo 'Keine Merkmale<br>';
132         $query = "SELECT_*_FROM_$tabelle.obj_stammdaten_WHERE
133         Land=_'$country'_$regionssuche_$citysuche
134         ORDER_BY_$tabelle.obj_stammdaten.ObjReferenz_LIMIT_$entries,$eintraege";
135         $query_all = "SELECT_*_FROM_$tabelle.obj_stammdaten_WHERE
136         Land=_'$country'_$regionssuche_$citysuche
137         ORDER_BY_$tabelle.obj_stammdaten.ObjReferenz_LIMIT_500";
138     }
139 }
140
141 /* Ermittlung aller Einträge und daraus Ableitung der Seitenanzahl */
142 if ($querymaxentries == true) {

```

```

143 $result_all = mysql_query($query_all) or die("Anfrage_gescheitert:" .
144     mysql_error());
145 $eintraege_all = mysql_num_rows($result_all);
146 $_SESSION['navigation'][3] = $eintraege_all;
147 $seiten = ceil($eintraege_all/$eintraege);
148 }
149 else {
150     $eintraege_all = $_SESSION['navigation'][3];
151     $seiten = ceil($eintraege_all/$eintraege);
152 }
153
154 $result = mysql_query($query) or die("Anfrage_gescheitert:" . mysql_error());
155
156 /* Warnung wenn mehr als 499 Einträge, Query ist auf 500 limitiert */
157 if ($eintraege_all >= 499) {
158     $info = '<div>Achtung, Ihre Anfrage brachte mehr als 500 Ergebnisse. Bitte
159     _schr&#228;nken_sie_die_Suche_ein!</div>';
160 }
161 elseif($eintraege_all < 1) {
162     $_SESSION['meldung'][0] = 'Ihre Suche f&#252;hrte leider zu keinem Ergebnis';
163     Header("Location:index.php?sid=$sessionid");
164 }
165 ?>
166
167 <!DOCTYPE html PUBLIC "-//W3C//DTD_XHTML1.0_Transitional//EN"
168     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
169 <html>
170     <html xmlns="http://www.w3.org/1999/xhtml">
171     <head>
172         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
173         <title>Ausgabe</title>
174         <link href="<?php_echo_$downward;?>css/ferienhaus_ibe.css" rel="stylesheet"
175             type="text/css">
176         <script type="text/javascript">
177             function XmlHttp( ){
178                 this.CreateXmlHttpRequest = CreateXmlHttpRequest;
179                 this.GetUrlContent = GetUrlContent;
180                 this.GetResponseText = GetResponseText;
181                 this.GetReadyState = GetReadyState;
182                 this.HttpMethod = 'GET'; // default
183                 this.objXmlHttp = this.CreateXmlHttpRequest();
184             }
185
186             // Initialize XMLHttpRequest
187             function CreateXmlHttpRequest(){
188                 var xmlhttp=false;
189                 try {
190                     xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
191                 } catch (e) {
192                     try {
193                         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
194                     } catch (E) {
195                         xmlhttp = false;
196                     }
197                 }
198
199                 if (!xmlhttp && typeof XMLHttpRequest!= 'undefined') {
200                     xmlhttp = new XMLHttpRequest();
201                 }
202                 return xmlhttp;
203             }
204
205             var objXMLHttp = new XmlHttp();
206             var objXMLHttp2 = new XmlHttp();
207             var objXMLHttp3 = new XmlHttp();
208             var objXMLHttp4 = new XmlHttp();

```

```

209 var objXMLHttp5 = new XmlHttp();
210
211 function GetReadyState( ){
212     return this.objXmlHttp.readyState;
213 }
214
215 function GetResponseText( ){
216     return this.objXmlHttp.responseText;
217 }
218
219 function GetUrlContent( strUrl , objOnReadyStateChangeFunction ){
220     this.objXmlHttp.open(this.HttpMethod, strUrl, true);
221     this.objXmlHttp.setRequestHeader( 'Content-Type', 'text/xml;_charset=UTF-8' );
222     if(objOnReadyStateChangeFunction){
223         this.objXmlHttp.onreadystatechange=function(){
224             objOnReadyStateChangeFunction();
225         }
226     }
227     this.objXmlHttp.send( null )
228 }
229
230 function GetResponse( ){
231     if (objXMLHttp.GetReadyState()==4) {
232         // save response in inner html of result object
233         var objMyDiv = document.getElementById( 'MyDiv' );
234         objMyDiv.innerHTML = objXMLHttp.GetResponseText( );
235     }
236 }
237
238 function GetResponse2( ){
239     if (objXMLHttp2.GetReadyState()==4) {
240         // save response in inner html of result object
241         var objMyDiv2 = document.getElementById( 'MyDiv2' );
242         objMyDiv2.innerHTML = objXMLHttp2.GetResponseText( );
243     }
244 }
245
246 function UpdateDiv( FieldValue ){
247     if( FieldValue != '' ){
248         var objMyDiv = document.getElementById( 'MyDiv' );
249         var objMyDiv2 = document.getElementById( 'MyDiv2' );
250         var objMyDiv3 = document.getElementById( 'MyDiv3' );
251         objXMLHttp.GetUrlContent( 'http://' + location.host +
252             '/ferienhaus-suche/ajax/liste.php?L1=' + FieldValue , GetResponse );
253     }
254     return;
255 }
256
257 function UpdateDiv2( FieldValue2, FieldValue1){
258     if( FieldValue2 != '' && FieldValue1 != '' ){
259         var objMyDiv2 = document.getElementById( 'MyDiv2' );
260         var objMyDiv3 = document.getElementById( 'MyDiv3' );
261         objMyDiv2.innerHTML = '<table><tr><td _class="geotd1">Ort:</td><td>
262         <select _class="geoselect" _name="L3" _size="1" _style="width:230px">
263         <option _value="kA">Bitte _Warten</option></select></td></tr></table>';
264         //objMyDiv3.innerHTML = '';
265         objXMLHttp2.GetUrlContent( 'http://' + location.host +
266             '/ferienhaus-suche/ajax/liste.php?L3=NULL&L2=' + FieldValue2 + '&L1=' +
267             FieldValue1+'&Level=2', GetResponse2 );
268     }
269     return;
270 }
271
272 </script>
273 <noscript>
274     <!-- Ihr Browser unterst#252;tzt kein Javascript. Leite auf Non Javascript
275     Version um<br> //-->
276     <meta http-equiv="refresh" content="0;_URL=uebersicht_ferienhaueser-nj.php">

```

```

275     </noscript>
276 </head>
277 <body>
278     <!-- ** Beginn Ausgabebereich** //-->
279     <!-- ***Linkleiste oben*** //-->
280     <div>
281         <?php
282             $seite = 'uebersicht';
283             include $pfad.'/include/suche_headlinks.php';
284         ?>
285     </div>
286
287     <!-- *Abfrage der Stammdaten aus Datenbank* //-->
288     <!-- ***Start Container 10x Hausanzeige (Hauptcontainer)*** //-->
289     <div id="main">
290         <div class="navibox">
291             <div style="text-align:right;_float:right;">
292                 Anzahl Suchergebnisse:
293                 <select name="liste" size="1" onchange="UpdateDiv(this.value)">
294                     <option value="<?php echo $eintraege;_?>" selected="selected"><?php
295                         echo $eintraege; ?></option>
296                     <option value="10"></option>
297                     <option value="10">10</option>
298                     <option value="20">20</option>
299                     <option value="50">50</option>
300                 </select>
301             </div>
302             <!-- ***Start Seitennavigation*** //-->
303             <?php
304                 $seitenzaehler = 1;
305                 //Start Container Seitennavigation
306                 echo 'Zur<#252;ck&nbsp;';
307                 for ($seiten_count = $seitenstart; $seiten_count <= $seiten; $seiten_count++)
308                 {
309                     if ($seitenzaehler <= 10) {
310                         //+10, da die Seitenzählung sonst mit 0 beginnen würde
311                         $aktuelleseite = ($sentries+10)/$eintraege;
312                         //Startdatensatz entspr Seite = (Seite n*einträge pro Seite)-10
313                         $eintragsstart = $seiten_count*$eintraege-10;
314                         //Standardwert wenn nicht zu nahe an den Rändern
315                         $seitenstart_half = $seiten_count-3;
316                         if ($seiten_count >= 1 AND $seiten_count <= 3) {
317                             $seitenstart_half = 1;
318                         }
319                         if ($seiten < 10) {
320                             if ($seiten_count >= 4 AND $seiten_count <= 9) {
321                                 $seitenstart_half = 1;
322                             }
323                         }
324                         if ($seiten >= 10) {
325                             //Folgende Einträge verschieben die Seitenabfrage
326                             //kontinuierlich nach hinten
327                             if ($seiten_count >= $seiten AND $seiten_count <= $seiten-5) {
328                                 $seitenstart_half = $seiten-9;
329                             }
330                         }
331                         if ($aktuelleseite == $seiten_count) {
332                             $getstring = '?seitenstart='.$seitenstart_half.
333                                 '&eintrakt='.$eintragsstart.'&sid='.$sessionid;
334                             echo '<a_style="color:#FF0000;"_href="uebersicht_ferienhaueser.php'.
335                                 $getstring.'">'. $seiten_count.'</a>_|_';
336                             $getstring = '';
337                         }
338                         else {
339                             $getstring = '?seitenstart='.$seitenstart_half.
340                                 '&eintrakt='.$eintragsstart.'&sid='.$sessionid;

```

```

341         echo ' <a href="uebersicht_ferienhaeuser.php'.
342             $getstring.'">' . $seiten_count . '</a>_|_';
343     }
344 }
345 $seitenzaehler++;
346 }
347 echo 'Vor';
348 ?>
349 </div>
350 <!-- ***Ende Seitennavigation*** //-->
351
352 <?php
353 echo $info;
354 while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
355     $objektrefcode = $line['ObjReferenz'];
356     $objektkategorie = $line['Kategorie'];
357     $anbieternr = $line['IDAnbieter'];
358     $objekt_land = $line['Land'];
359     $objekt_region = $line['Region'];
360     $objekt_ort = $line['Ort'];
361
362     $objektdaten = new datensaetze;
363
364     $objektdaten->land_holen($objekt_land);
365     $objektdaten->region_holen($objekt_region);
366     $objektdaten->ort_holen($objekt_ort);
367     $objektdaten->haustext_holen($objektrefcode);
368     $objektdaten->merkmale_holen($objektrefcode, $objektkategorie);
369     $objektdaten->hausbilder_holen($objektrefcode);
370
371     $objekt_land_name = $objektdaten->fh_land;
372     $objekt_region_name = $objektdaten->fh_region;
373     $objekt_ort_name = $objektdaten->fh_ort;
374     $beschreibungstext_haus = $objektdaten->desc_haus;
375     $mm_merkmale = $objektdaten->merkmale_array;
376     $mm_anzahl = $objektdaten->anzahl_array;
377     $mm_entfernung = $objektdaten->entfernung_array;
378     $mm_buchopt = $objektdaten->buchoption_array;
379     $hausbild = $objektdaten->hausbild_einzelbild;
380 ?>
381 <!-- *Ausgabe der Ferienobjekte* //-->
382 <!-- ***Container Haus Anfang*** //-->
383 <div class="ueb_hausbox">
384
385     <!-- *Abruf Beschreibung Häuser, Lage* //-->
386     <div name="beschreibung" class="ueb_beschrtext">
387         <!-- Ort, Region ausgeben //-->
388         <h1><?php echo $objekt_region_name.' , '. $objekt_ort_name; ?></h1>
389         <!-- Wenn verfügbar Hausname ausgeben //-->
390         <?php echo $beschreibungstext_haus; ?><a class="links"
391             href="details_ferienhaus.php?oref=<?php echo
392             $objektrefcode;?>&okat=<?php echo $objektkategorie;?>&date=
393             <?php echo $vakanzabfrage_datum;?>&days=<?php echo
394             $aufenthaltsdauer;?>&anbieter=<?php echo $anbieternr;
395             <?php echo $view=1&sid=<?php echo $sessionid;?>">mehr</a>
396     </div>
397
398     <!-- *Abruf Merkmale aus Session Array, Ausgabe* //-->
399     <div name="merkmale" class="ueb_merkmale">
400         <?php
401         if ($mm_merkmale != '') {
402             foreach ($mm_merkmale AS $merkmal) {
403                 if ($merkmal != '') {
404                     echo $merkmal.'<br>';
405                 }
406             }

```

```

407     }
408     if ($mm_anzahl != '') {
409         foreach ($mm_anzahl AS $anzahl) {
410             if ($anzahl != '') {
411                 echo $anzahl.'<br>';
412             }
413         }
414     }
415     if ($mm_entfernung != '') {
416         foreach ($mm_entfernung AS $entfernung) {
417             if ($entfernung != '') {
418                 echo $entfernung.'<br>';
419             }
420         }
421     }
422     if ($mm_buchopt != '') {
423         foreach ($mm_buchopt AS $buchoption) {
424             if ($buchoption != '') {
425                 echo $buchoption.'<br>';
426             }
427         }
428     }
429     ?>
430 </div>
431
432 <!-- *Abruf Preise Häuser* //-->
433 <div name="preise" class="ueb_preise">
434     <!-- Preisangabe sehr unterschiedlich, nicht wochengebundene Abfrage mit
435     Eingrenzung einer Woche deshalb kritisch //-->
436     <?php
437         if ($vakanzabfrage_datum != '') {
438             $holen_preise = "SELECT Preis FROM obj_preise WHERE ObjReferenz =
439             '$_OBJKTRFCODE' AND Kategorie = '$_OBJKTKATEGORIE' AND Aufenthalt =
440             '$_AUFENTHALTSDAUER' AND Datumvon <= '$_$vakanzabfrage_datum' LIMIT 1";
441             $abfragen_preise = mysql_query($holen_preise);
442             if (mysql_num_rows($abfragen_preise) < 1) {
443                 $preisaktuell = 'k.A.';
444             }
445             else {
446                 while($line_preise = mysql_fetch_array($abfragen_preise,
447                 MYSQL_ASSOC)) {
448                     $preisaktuell = $line_preise['Preis'];
449                 }
450             }
451             echo 'ab: '.$vakanzabfrage_datum.'<br>';
452             echo $aufenthaltsdauer.' Tage<br>';
453             if ($preisaktuell == 'k.A.') {
454                 echo 'Preis: '.$preisaktuell.'<br>';
455             }
456             else {
457                 echo 'Preis: '.$preisaktuell.'&#128;<br>';
458             }
459         }
460         ?>
461     </div>
462
463 <!-- *Abruf Bilder Häuser* //-->
464 <div name="bilder" class="ueb_bilder">
465     <?php
466         $bildgr = getimagesize($hausbild);
467         ?>
468     <!-- <a class="links" href="details_ferienhaus.php?oref=<?php echo
469     $_OBJKTRFCODE;?>&okat=<?php echo $_OBJKTKATEGORIE;?>&date=<?php echo
470     $_$vakanzabfrage_datum;?>&days=<?php echo
471     $_$aufenthaltsdauer;?>&anbieter=<?php echo $_$anbieternr;
472     $_$?>&view=1">mehr</a> //-->

```



```

473         <?php
474         if (is_array($bildgr)) {
475             ?>
476             <a class="links" href="details_ferienhaus.php?oref=<?php_echo
477 .....$objektrefcode;?>&okat=<?php_echo_.$objektkategorie;?>&date=<?php_echo
478 .....$vakanzabfrage_datum;?>&days=<?php_echo
479 .....$aufenthaltsdauer;?>&anbieter=<?php_echo_.$anbieternr;
480 .....?>&view=1&sid=<?php_echo_.$sessionid;?>"></a><br>
484         <?php
485         }
486         else {
487             ?>
488             <a class="links" href="details_ferienhaus.php?oref=<?php_echo
489 .....$objektrefcode;?>&okat=<?php_echo_.$objektkategorie;?>&date=
490 .....<?php_echo_.$vakanzabfrage_datum;?>&days=<?php_echo
491 .....$aufenthaltsdauer;?>&anbieter=<?php_echo_.$anbieternr;_?>&view=1">
492 .....
495 .....</a><br>
496         <?php
497         }
498         ?>
499         <!-- Objekt Referenzcode //-->
500         <?php echo $line['ObjReferenz'].'.'.'. $line['Kategorie']; ?>
501     </div>
502     <!-- ***Container Haus Ende*** //-->
503 </div>
504 <?php
505 }
506 ?>
507 <!-- Ende Container 10x Hausanzeige //-->
508 </div>
509
510 <?php
511 mysql_free_result($result);
512 mysql_close($db_open);
513 ?>
514 </body>
515 </html>

```

G.3. Ferienhaus Detailansicht

Listing G.3: Quelltext Ferienhaus Detailansicht

```

1 <?php
2 $sessionid = $_REQUEST['sid'];
3 if (!isset($_COOKIE['PHPSESSID'])) {
4     session_id($sessionid);
5 }
6 session_start();
7 echo session_id(). '<br>';
8 echo $sessionid;
9 ?>
10 <noscript>
11     <!-- Ihr Browser unterst#252;tzt kein Javascript.
12     Leite auf Non Javascript Version um<br> //-->
13     <meta http-equiv="refresh" content="0;URL=details_ferienhaus-nj.php">
14 </noscript>
15 <?php
16 /* ----- Verzeichnisse ermitteln, Klassen und Datenbankzugriff einbinden ----- */

```

```

17 $pfad = getenv("DOCUMENT_ROOT");
18 /* *Tiefe zum Stammverzeichnis ermitteln* */
19 $downward = "";
20 $Pfad = getenv("REQUEST_URI");
21 $Pfadteile = explode("/", $Pfad);
22 for($i=count($Pfadteile)-1;$i>=1;$i--)
23 {
24     $downward = $downward . "../";
25 }
26
27 $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
28 $inc_objektklasse = $pfad.'/include/class-objekte-db.php';
29 include $datenbankzugriff;
30 include $inc_objektklasse;
31
32 /* **Einbinden der Check und Initialisierungsroutinen**
33     **Für übergebene Daten von der uebersicht-ferienhauser.php ** */
34 require_once("varcheck2.php");
35
36 /* —— Werte holen —— */
37 $objektreferenz = $_SESSION['objekt'][0];
38 $objektkategorie = $_SESSION['objekt'][1];
39 $aufenthalt_start = $_SESSION['objekt'][2];
40 $aufenthalt_dauer = $_SESSION['objekt'][3];
41 $anbieter = $_SESSION['objekt'][4];
42
43 /* Notwendige Datensätze für die Detailansicht abrufen, über "datensaetze" Klasse */
44 $objektdaten = new datensaetze;
45 /* —— Land-, Region-, Ort-ID ermitteln —— */
46 $objektdaten->stammdaten_holen($objektreferenz, $objektkategorie);
47 $objekt_land = $objektdaten->stammdaten_land;
48 $objekt_region = $objektdaten->stammdaten_region;
49 $objekt_ort = $objektdaten->stammdaten_ort;
50
51 /* —— Zugehöriges Land, Region und Ort ermitteln —— */
52 $objektdaten->land_holen($objekt_land);
53 $name_land = $objektdaten->fh_land;
54 $objektdaten->region_holen($objekt_region);
55 $name_region = $objektdaten->fh_region;
56 $objektdaten->ort_holen($objekt_ort);
57 $name_ort = $objektdaten->fh_ort;
58
59 /* —— Sämtliche dem Objekt zugehörige Bilder ermitteln —— */
60 $objektdaten->hausbilder_holen($objektreferenz);
61 $bilder = $objektdaten->hausbilder;
62 $bilder_nummer = $objektdaten->hausbilder_nummer;
63 $hausbilder_anzahl = count($bilder);
64 $i = $hausbilder_anzahl;
65
66 $objektdaten->wohnungsbilder_holen($objektreferenz, $objektkategorie);
67 $wohnungsbilder_nummer = $objektdaten->wohnungsbilder_nummer;
68 $k = 0;
69 if ($objektdaten->wohnungsbilder != '') {
70     foreach ($objektdaten->wohnungsbilder AS $wohnungsbilder) {
71         $bilder[$i] = $wohnungsbilder;
72         $bilder_nummer[$i] = $wohnungsbilder_nummer[$k];
73         $i++;
74         $k++;
75     }
76 }
77 $allebilder_anzahl = count($bilder);
78
79 /* —— Beschreibungstexte holen —— */
80 $objektdaten->haustext_holen($objektreferenz);
81 $haustext = $objektdaten->desc_haus_komplett;
82 $objektdaten->wohnungstext_holen($objektreferenz, $objektkategorie);

```

```

83 $wohnungstext = $objektdaten->desc-wohnung-komplett;
84
85 /* —— Merkmale holen —— */
86 $objektdaten->merkmale.holen($objektreferenz,$objektkategorie);
87
88 /* —— Hausname holen —— */
89 $objektdaten->hausname.holen($objektreferenz,$objektkategorie);
90 $objektname = $objektdaten->objektname;
91 /* —— Bewertung holen —— */
92 $objektdaten->quali.holen($objektreferenz,$objektkategorie);
93 $hausbewertung = $objektdaten->objektquali;
94
95 /* *Entspr Header für die Links, oben setzen* */
96 if (isset($_GET['direkt'])) {
97     if ($_GET['direkt'] == 1) {
98         $seite = 'detailansichtdirekt';
99     }
100     else {
101         $seite = 'detailansicht';
102     }
103 }
104 else {
105     $seite = 'detailansicht';
106 }
107 ?>
108 <html>
109 <html xmlns="http://www.w3.org/1999/xhtml">
110 <head>
111 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
112 <title>Abfragetests</title>
113 <link href="../css/ferienhaus_ibe.css" rel="stylesheet" type="text/css">
114 <script type="text/javascript">
115 function XmlHttp( ){
116     this.CreateXmlHttpRequest = CreateXmlHttpRequest;
117     this.GetUrlContent           = GetUrlContent;
118     this.GetResponseText        = GetResponseText;
119     this.GetReadyState           = GetReadyState;
120     this.HttpMethod = 'GET'; // default
121     this.objXmlHttp = this.CreateXmlHttpRequest();
122 }
123
124 // Initialize XMLHttpRequest
125 function CreateXmlHttpRequest(){
126     var xmlhttp=false;
127     try {
128         xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
129     } catch (e) {
130         try {
131             xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
132         } catch (E) {
133             xmlhttp = false;
134         }
135     }
136
137     if (!xmlhttp && typeof XMLHttpRequest!= 'undefined') {
138         xmlhttp = new XMLHttpRequest();
139     }
140     return xmlhttp;
141 }
142
143 var objXMLHttp = new XmlHttp();
144
145 function GetReadyState( ){
146     return this.objXmlHttp.readyState;
147 }
148

```

```

149     function GetResponseText( ){
150         return this.objXmlHttp.responseText;
151     }
152
153     function GetUrlContent( strUrl , objOnReadyStateChangeFunction ){
154         this.objXmlHttp.open( this.HttpMethod , strUrl , true );
155         this.objXmlHttp.setRequestHeader( 'Content-Type' , 'text/xml;_charset=UTF-8' );
156         if( objOnReadyStateChangeFunction ){
157             this.objXmlHttp.onreadystatechange=function(){
158                 objOnReadyStateChangeFunction();
159             }
160         }
161         this.objXmlHttp.send( null )
162     }
163
164     //This function is called when we get the data back from the server.
165     function GetResponse( ){
166         if (objXMLHttp.GetReadyState()==4) {
167             // save response in inner html of result object
168             var objBildDiv = document.getElementById( 'BildDiv' );
169             objBildDiv.innerHTML = objXMLHttp.GetResponseText( );
170         }
171     }
172
173     function UpdateDiv( FieldValue ){
174         if( FieldValue != '' ){
175             var objBildDiv = document.getElementById( 'BildDiv' );
176             var objBildDiv2 = document.getElementById( 'BildDiv2' );
177             objBildDiv.innerHTML = '<img_style="width:320px;_height:240px;"
178             .....src="ibelogo.jpg"_width="300px"_height="210px">';
179             objXMLHttp.GetUrlContent( 'http://' + location.host +
180                 '/ferienhaus-suche/ajax/detailseite-bildanzeige.php?L1=' +
181                 FieldValue , GetResponse );
182         }
183         return;
184     }
185 </script>
186 </head>
187 <body>
188 <div>
189     <!-- *Linkleiste includieren* -->
190     <?php
191         // $seite = 'detailansicht';
192         include $pfad.'/include/suche_headlinks.php';
193     ?>
194 </div>
195
196 <!-- **Hauptcontainer Start** -->
197 <div id="main">
198     <!-- ***Start Geo u. Objektcodecontainer*** -->
199     <div class="det_geobox">
200         <div class="det_objcode">
201             <!-- *Ausgabe Ort, Region, Land, Objektcode* -->
202             Ferienhaus in <?php echo $name_ort; ?><br><?php echo $name_region.' ,
203             .....$name_land; ?><br><br>
204             Objekt:<h1><?php echo $objektname; ?><h1><?php echo
205             $objektreferenz.'.'. $objektkategorie; ?></h1>
206         </div>
207         <div class="det_link1">
208             <a href="merklste.php?aktion=merken&sid=<?php echo $_sessionid;
209             .....?>">Merken</a>
210         </div>
211         <div class="det_link2">
212             <a href="vakanzen.php?sid=<?php echo $_sessionid; ?>">Buchen</a>
213         </div>
214         <div class="det_link3">

```

```

215         <a class="links" href="vakanzen.php?sid=<?php_echo_$sessionId;
216         .....?>">Vakanzen</a>
217     </div>
218     <div class="det_info">
219         <?php
220             echo $_SESSION['error'];
221             $_SESSION['error'] = '';
222         ?>
223     </div>
224 </div>
225 <!-- ***Ende Geo u. Objektcodecontainer*** //-->
226
227 <!-- ***Start Bildcontainer*** //-->
228 <div NAME="BildDiv" ID="BildDiv" class="det_bildbox">
229     <?php
230         $bildloopen = getimagesize($bilder[0]);
231         if (is_array($bildloopen)) {
232             echo '<img_style="width:320px;_height:240px;"
233 .....src="'. $downward.'include/grafik.php?pic='.$bilder[0]. '&cx=320&cy=240" _/>';
234         }
235         else {
236             echo '<img_style="width:320px;_height:240px;"
237 .....src="'. $downward.'include/grafik.php?pic='
238             $downward.'ferienhaus-suche/ibelogo.jpg&cx=320&cy=240" _/>';
239         }
240     ?>
241 <!-- ***Ende Bildcontainer*** //-->
242 </div>
243
244 <!-- ***Start Merkmalscontainer*** //-->
245 <div class="det_merkmalsbox">
246     <!-- *Merkmale ausgeben* //-->
247     <div class="det_merkmale">
248         <?php
249             if ($objektdaten->merkmale_array != '') {
250                 echo '<p><h1>Merkmale</h1></p>';
251                 foreach ($objektdaten->merkmale_array AS $merkmal) {
252                     if ($merkmal != '') {
253                         echo $merkmal.'<br>';
254                     }
255                 }
256             }
257             if (is_array($hausbewertung)) {
258                 echo '<h1>Bewertung</h1>';
259                 if ($hausbewertung['Quali'] != 0) {
260                     echo 'Qualit&#228;t:'. $hausbewertung['Quali']. '<br>';
261                 }
262                 if ($hausbewertung['CevalA'] != 0) {
263                     echo 'Lage, _Aussicht:'. $hausbewertung['CevalA']. '<br>';
264                 }
265                 if ($hausbewertung['CevalB'] != 0) {
266                     echo 'Aussenbereich:'. $hausbewertung['CevalB']. '<br>';
267                 }
268                 if ($hausbewertung['CevalC'] != 0) {
269                     echo 'Einrichtung:'. $hausbewertung['CevalC']. '<br>';
270                 }
271                 if ($hausbewertung['CevalD'] != 0) {
272                     echo 'Ruhe _der _Lage:'. $hausbewertung['CevalD']. '<br>';
273                 }
274                 if ($hausbewertung['CevalE'] != 0) {
275                     echo 'K&#252;che:'. $hausbewertung['CevalE']. '<br>';
276                 }
277                 if ($hausbewertung['CevalF'] != 0) {
278                     echo 'Zufahrt:'. $hausbewertung['CevalF'];
279                 }
280             }

```

```

281     ?>
282 </div>
283 <div class="det_zimmer">
284 <!-- *Anzahl Zimmern, Schlafzimmer, Personen u.s.w. ausgeben* -->
285 <?php
286     if ($objektdaten->anzahl_array != '') {
287         echo '<p><h1>R&#228;umlichkeiten</h1></p>';
288         foreach ($objektdaten->anzahl_array AS $anzahl) {
289             if ($anzahl != '') {
290                 echo $anzahl.'<br>';
291             }
292         }
293     }
294     ?>
295 <!-- *Ausgabe Entfernung zu wichtigen Plätzen* -->
296 <?php
297     if ($objektdaten->entfernung_array != '') {
298         echo '<p><h1>Entfernungen</h1></p>';
299         foreach ($objektdaten->entfernung_array AS $entfernung) {
300             if ($entfernung != '') {
301                 echo $entfernung.'<br>';
302             }
303         }
304     }
305     ?>
306 </div>
307 <!-- ***Ende Merkmalscontainer*** -->
308 </div>
309
310 <!-- Ausgabe einer Vorschau-gallerie mit max 16 Bildern -->
311 <!-- ***Bildwahlcontainer start*** -->
312 <div NAME="BildDiv2" ID="BildDiv2" class="det_bildwbox">
313 <?php
314     /* init */
315     $k=-1; $m = 0;
316     $bilderhash = array(0);
317     /* alle Bilder durchgehen */
318     for ($i=0; $i < 16; $i++) {
319         /* Bild vorhanden? */
320         $bildopen = @getimagesize($bilder[$i]);
321         if (is_array($bildopen)) {
322             /* Hash aktuelles Bild */
323             $hash1 = sha1_file($bilder[$i]);
324             $bildausgabe = 1;
325             /* Hash bereits im array? */
326             foreach ($bilderhash AS $hash) {
327                 if ($hash == $hash1) {
328                     $bildausgabe = 0;
329                 }
330             }
331             /* Ablage, Hash aktuelles Bild, im Array */
332             $bilderhash[$m] = $hash1;
333             $m++;
334
335             if ($bildausgabe == 1) {
336                 echo '';
339                 $k++;
340             }
341             if ($k == 3) {
342                 echo '<br>';
343                 $k=-1;
344             }
345         }
346     }

```

```

347     ?>
348     <!-- ***Bildwahlcontainer ende*** //-->
349 </div>
350
351 <!-- *Ausgabe der Beschreibungstexte* //-->
352 <!-- ***Beschreibungscontainer Start*** //-->
353 <div NAME="Beschreibungstexte" ID="Beschreibungstexte" class="det_text">
354     <?php
355         if (strlen($haustext) > 5) {
356             echo '<p><h1>Das_Haus:</h1></p><p>'. $haustext. '</p>';
357         }
358         if (strlen($wohnungstext) > 5) {
359             echo '<p><h1>Die_Wohnung:</h1></p><p>'. $wohnungstext. '</p>';
360         }
361     ?>
362     <!-- ***Ende Beschreibungscontainer*** //-->
363 </div>
364 <!-- Hauptcontainer Ende //-->
365 </div>
366 </body>
367 </html>

```

G.4. Ferienhaus Merkliste

Listing G.4: Quelltext Ferienhaus Merkliste

```

1 <?php
2 $sessionid = $_REQUEST['sid'];
3 if (!isset($_COOKIE['PHPSESSID'])) {
4     session_id($sessionid);
5 }
6 session_start();
7 $objektreferenz = $_SESSION['objekt'][0];
8 $objektkategorie = $_SESSION['objekt'][1];
9 $anbieter = $_SESSION['objekt'][4];
10
11 /* Datums- und Personendaten */
12 $vakanzabfrage_datum = $_SESSION['allgemein'][1][0];
13 $aufenthaltsdauer = $_SESSION['allgemein'][1][1];
14
15 if (isset($_GET['aktion']) ) {
16     if ($_GET['aktion'] == 'merken') {
17         $aktion = 'merken';
18     }
19     elseif ($_GET['aktion'] == 'auslesen') {
20         $aktion = 'auslesen';
21     }
22     elseif ($_GET['aktion'] == 'loeschen') {
23         $aktion = 'loeschen';
24     }
25     elseif ($_GET['aktion'] == 'allesloeschen') {
26         $aktion = 'allesloeschen';
27     }
28     else {
29         $aktion = 'auslesen';
30     }
31 }
32 else {
33     $aktion = 'auslesen';
34 }
35
36 $objektnummer=$objektreferenz.'_'.$objektkategorie.'_'.$anbieter;
37 $pfad = getenv("DOCUMENT_ROOT");
38 $downward = "";

```

```

39 $Pfad = getenv("REQUEST_URI");
40 $Pfadteile = explode("/", $Pfad);
41 for ($i=count($Pfadteile)-1; $i>=1; $i--)
42 {
43     $downward = $downward . " ../";
44 }
45
46 $datenbankzugriff = $Pfad.' /include/opendb-ferienhaus.php';
47 $inc_objektklasse = $Pfad.' /include/class_objekte-db.php';
48 include $datenbankzugriff;
49 include $inc_objektklasse;
50
51 class merkliste
52 {
53     var $objekt_anbieter;
54     var $objekt_ref;
55     var $objekt_kat;
56
57     function eintragen($objektnummer1) {
58         $eintrag = 1;
59         for ($i=0; $i < 25; $i++) {
60             if (strlen($_COOKIE['merken'][$i]) < 5 OR $_COOKIE['merken'][$i] == '0') {
61                 if ($eintrag != 0) {
62                     setcookie("merken[$i]", $objektnummer1, time()+1209600);
63                     $_SESSION['error'] = "Objekt_wurde_in_die_Merkliste_&#252;bernommen";
64                     $eintrag = 0;
65                 }
66             }
67
68             if ($_COOKIE['merken'][$i] == $objektnummer1) {
69                 $_SESSION['error'] = "Dieses_Objekt_befindet_sich_bereits_in_der_Merkliste";
70                 $eintrag = 0;
71             }
72         }
73     }
74
75     function auslesen() {
76         for ($i=0; $i < 25; $i++) {
77             if (!strlen($_COOKIE['merken'][$i]) < 5 OR $_COOKIE['merken'][$i] != 0) {
78                 $objektnummer_array = explode("_", $_COOKIE['merken'][$i]);
79                 $this->objekt_ref[$i] = $objektnummer_array[0];
80                 $this->objekt_kat[$i] = $objektnummer_array[1];
81                 $this->objekt_anbieter[$i] = $objektnummer_array[2];
82                 //print_r($objektnummer_array);
83             }
84         }
85     }
86
87     function loeschen($objektnummer1) {
88         for ($i=0; $i < 25; $i++) {
89             //echo $objektnummer1;
90             if ($_COOKIE['merken'][$i] == $objektnummer1) {
91                 setcookie("merken[$i]", 0, time()+1209600);
92             }
93         }
94     }
95
96     function allesloeschen() {
97         for ($i=0; $i < 25; $i++) {
98             setcookie("merken[$i]", 0, time()+1209600);
99         }
100     }
101 }
102
103 $merkfunktion = new merkliste;
104 if ($aktion == 'merken') {

```



```

105     $merkfunktion->eintragen($objektnummer);
106     Header("Location: details_ferienhaus.php?sid=$sessionid");
107     exit();
108 }
109
110 if ($aktion == 'loeschen') {
111     for ($i=0; $i < 25; $i++) {
112         $objektnummer = $_GET[$i];
113         $merkfunktion->loeschen($objektnummer);
114     }
115     Header("Location: merklste.php?aktion=auslesen?sid=$sessionid");
116 }
117
118 if ($aktion == 'allesloeschen') {
119     $merkfunktion->allesloeschen();
120     Header("Location: merklste.php?aktion=auslesen?sid=$sessionid");
121 }
122 ?>
123 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
124 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
125 <html>
126     <html xmlns="http://www.w3.org/1999/xhtml">
127     <head>
128         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
129         <title>Ausgabe</title>
130         <link href="../css/ferienhaus_ibe.css" rel="stylesheet" type="text/css">
131     </head>
132     <body>
133         <!-- ***Start Container 10x Hausanzeige*** -->
134
135         <?php
136             echo session_id(). '—<br>';
137             echo $sessionid;
138             if ($aktion == 'auslesen') {
139                 ?>
140                 <div>
141                     <?php
142                         $seite = 'merklste';
143                         include '../include/suche_headlinks.php';
144                     ?>
145                 </div>
146
147                 <div id="main">
148                     <form action="merklste.php" method="GET">
149                         <?php
150                             $objektdaten = new datensaetze;
151                             $merkfunktion->auslesen();
152
153                             for ($i_haus = 0; $i_haus < 25; $i_haus++) {
154                                 $db_objreferenz = $merkfunktion->objekt_ref[$i_haus];
155                                 $db_objkat = $merkfunktion->objekt_kat[$i_haus];
156                                 $db_objanbieter = $merkfunktion->objekt_anbieter[$i_haus];
157                                 $objektnummer= $db_objreferenz.'_'.$db_objkat.'_'.$db_objanbieter;
158
159                                 if (($db_objreferenz != 0 OR $db_objreferenz != '') AND
160                                     ($db_objkat != 0 OR $db_objkat != '')) {
161                                     ?>
162                                     <!-- ***Start Container 1 Haus*** -->
163                                     <div class="ueb_hausbox">
164                                         <?php
165
166                                             $objektdaten->stammdaten_holen($db_objreferenz,$db_objkat);
167
168                                             $regionscode = $objektdaten->stammdaten_region;
169                                             $sortscode = $objektdaten->stammdaten_ort;
170

```

```

171     $objektdaten->region_holen($regionscode);
172     $objektdaten->ort_holen($ortscode);
173     $objektdaten->haustext_holen($db_objreferenz);
174     $objektdaten->merkmale_holen($db_objreferenz,$db_objkat);
175     $objektdaten->hausbilder_holen($db_objreferenz);
176
177     ?>
178     <!-- *Beschreibung Häuser, Lage* //-->
179     <div name="beschreibung" class="ueb_beschrtext">
180         <h1><?php echo $objektdaten->fh_region.','.'
181             $objektdaten->fh_ort; ?></h1>
182         <?php echo $objektdaten->desc_haus; ?>
183         <?php echo $beschreibungstext_haus; ?><a class="links"
184             href="details_ferienhaus.php?oref=<?php
185             echo $db_objreferenz;?>&okat=<?php echo $db_objkat;?>&
186             anbieter=<?php echo $db_objanbieter;?>&view=1&sid=<?php
187             echo $sessionid; ?>">mehr</a>
188     </div>
189
190     <!-- *Merkmale* //-->
191     <div name="merkmale" class="ueb_merkmale">
192         <?php
193
194         if ($objektdaten->merkmale_array != '') {
195             foreach ($objektdaten->merkmale_array AS $merkmal) {
196                 if ($merkmal != '') {
197                     echo $merkmal.'<br>';
198                 }
199             }
200         }
201         if ($objektdaten->anzahl_array != '') {
202             foreach ($objektdaten->anzahl_array AS $anzahl) {
203                 if ($anzahl != '') {
204                     echo $anzahl.'<br>';
205                 }
206             }
207         }
208         if ($objektdaten->entfernung_array != '') {
209             foreach ($objektdaten->entfernung_array AS $entfernung) {
210                 if ($entfernung != '') {
211                     echo $entfernung.'<br>';
212                 }
213             }
214         }
215     ?>
216 </div>
217
218     <!-- *Preise zum ausgewählten Datum* //-->
219     <div name="preise" class="ueb_preise">
220         <!-- Preisangabe sehr unterschiedlich, nicht wochengebunden
221         Abfrage mit Eingrenzung einer Woche deshalb kritisch //-->
222         <?php
223         if ($vakanzabfrage_datum != '') {
224             $holen_preise = "SELECT Preis FROM obj_preise WHERE
225             ObjReferenz = '$objektreferenz' AND
226             Kategorie = '$objektkategorie' AND
227             Aufenthalt = '$aufenthaltsdauer' AND
228             Datumvon <= '$vakanzabfrage_datum' _LIMIT 1";
229             $abfragen_preise = mysql_query($holen_preise);
230             if (mysql_num_rows($abfragen_preise) < 1) {
231                 $preisaktuell = 'k.A.';
232             }
233         }
234         else {
235             while($line_preise = mysql_fetch_array(
236                 $abfragen_preise, MYSQL_ASSOC)) {
237                 $preisaktuell = $line_preise['Preis'];

```

```

237     }
238     }
239     echo 'ab_'. $vakanzabfrage_datum. '<br>';
240     echo $aufenthaltsdauer. '_Tage<br>';
241     if ($preisaktuell == 'k.A.') {
242         echo 'Preis: _'. $preisaktuell. '<br>';
243     }
244     else {
245         echo 'Preis: _'. $preisaktuell. '&#128;<br>';
246     }
247 }
248 ?>
249 </div>
250
251 <!-- *Bilder Hausansicht* -->
252 <div name="bilder" class="ueb_bilder">
253     <?php echo $beschreibungstext_haus; ?>
254     <a class="links" href="details_ferienhaus.php?oref=<?php
255     echo _$db_objreferenz;?>&okat=<?php echo _$db_objkat;?>&
256     anbieter=<?php echo _$db_objanbieter;?>&view=1&sid=<?php
257     echo _$sessionid;_?>"><img width="120" height="100"
258     src=<?php echo _$downward. 'include/grafik.php?pic=' .
259     $objektdaten->hausbild_einzelbild. '&x=120&y=100';_?>" /></a>
260     <!-- <img width="120" height="100" src=<?php
261     echo _$objektdaten->hausbild_einzelbild;_?>" /><br> -->
262 </div>
263
264 <div>
265     <input type="checkbox" name="<?php echo _$i_haus;_?>"
266     value="<?php echo _$objektnummer;_?>"></input>
267 </div>
268 <!-- ***Ende Container 1 Haus*** -->
269 </div>
270 <?php
271 }
272 }
273 ?>
274 <select name="aktion">
275     <option value="loeschen">Ausgew&#228;hlte l&#246;schen</option>
276     <option value="allesloeschen">Alle l&#246;schen</option>
277 </select>
278 <input name="sid" type="hidden" value="<?php echo _$sessionid;?>"
279 <input type="submit" value="Best&#228;tigen"></input>
280 </form>
281 </div>
282 <?php
283 }
284 ?>
285 <!-- ***Ende Container 10x Hausanzeige*** -->
286
287 </body>
288 </html>

```

G.5. Belegungsübersicht

Listing G.5: Quelltext Belegungsübersicht

```

1 <?php
2 $sessionid = $_REQUEST['sid'];
3 if (!isset($_COOKIE['PHPSESSID'])) {
4     session_id($sessionid);
5 }
6 session_start();
7 echo session_id(). '<br>';

```

```

8 echo $sessionid;
9 ?>
10 <noscript>
11 <!-- Ihr Browser unterst&#252;tzt kein Javascript. Leite auf
12 Non Javascript Version um<br> //-->
13 <meta http-equiv="refresh" content="0;_URL=vakanzen-nj.php">
14 </noscript>
15 <?php
16 $objektreferenz = $_SESSION['objekt'][0];
17 $objektkategorie = $_SESSION['objekt'][1];
18 $aufenthalt_start = $_SESSION['objekt'][2];
19 $anbieter = $_SESSION['objekt'][4];
20 $aufenthalt_dauer = $_SESSION['objekt'][3];
21 $_SESSION['objekt'][5] = '';
22 if ($aufenthalt_dauer == '') {
23     $aufenthalt_dauer = 7;
24 }
25 /* **Einbinden der Check und Initialisierungsroutinen**
26     **Für übergebene Daten ** */
27 require_once("varcheck2.php");
28 $pfad = getenv("DOCUMENTROOT");
29 $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
30 include $datenbankzugriff;
31 $inc_objektklasse = $pfad.'/include/class_objekte-db.php';
32 include $inc_objektklasse;
33 $inc_datumsklasse = $pfad.'/include/class_datumfkt.php';
34 include $inc_datumsklasse;
35 $inc_geldklasse = $pfad.'/include/class_geld.php';
36 include $inc_geldklasse;
37
38 $interhome_vakanz_live = $pfad.'/ferienhaus-suche/live/interhome-vakanzausgabe.php';
39 $wolters_vakanz_live = $pfad.'/ferienhaus-suche/live/wolters-vakanzausgabe.php';
40 $interhome_price_live = $pfad.'/ferienhaus-suche/live/interhome-preisausgabe.php';
41 $wolters_price_live = $pfad.'/ferienhaus-suche/live/wolters-preisausgabe.php';
42 $onfh_vakanz_live = $pfad.'/ferienhaus-suche/live/onfh-vakanzausgabe.php';
43
44 $objektdaten = new datensaetze;
45 $datumfkt = new zeitfunktionen;
46 $moneur = new changeto-eur;
47
48 if ($aufenthalt_start == '' OR $aufenthalt_start == '..') {
49     $aufenthalt_start = date("d.m.Y");
50 }
51 /* *Zusatzleistungen ermitteln* */
52 $objektdaten->zusatzleistungen-holen($anbieter, $objektreferenz, $objektkategorie);
53 $zubs = $objektdaten->zubs;
54 $anzahl_zubs = count($zubs)-1;
55
56 $datum_aktuell = explode(".", date("m.Y"));
57 $month_akt = $datum_aktuell[0]+0;
58 $datum_array = array(1 => 'Januar', 2 => 'Februar', 3=> 'M&#228;rz', 4=> 'April',
59 5 => 'Mai', 6 => 'Juni', 7 => 'Juli', 8 => 'August', 9 => 'September',
60 10 => 'Oktober', 11 => 'November', 12 => 'Dezember');
61
62 $downward = "";
63 $Pfad = getenv("REQUEST_URI");
64 $Pfadteile = explode("/", $Pfad);
65 for($i=count($Pfadteile)-1;$i>=1;$i--)
66 {
67     $downward = $downward . "../";
68 }
69 ?>
70 <!DOCTYPE html PUBLIC "-//W3C//DTD_XHTML_1.0_Transitional//EN"
71 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
72 <html>
73 <html xmlns="http://www.w3.org/1999/xhtml">

```

```

74 <head>
75 <meta http-equiv="Content-Type" content="text/html;_charset=UTF-8">
76 <title>Abfragetests</title>
77 <link href="<?php_echo_$downward?>css/ferienhaus_ibe.css"
78 rel="stylesheet" type="text/css">
79 <script type="text/javascript">
80 function XmlHttp( ){
81     this.CreateXmlHttpRequest = CreateXmlHttpRequest;
82     this.GetUrlContent          = GetUrlContent;
83     this.GetResponseText       = GetResponseText;
84     this.GetReadyState         = GetReadyState;
85     this.HttpMethod = 'GET'; // default
86     this.objXmlHttp = this.CreateXmlHttpRequest();
87 }
88
89 // Initialize XMLHttpRequest
90 function CreateXmlHttpRequest(){
91     var xmlhttp=false;
92     try {
93         xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
94     } catch (e) {
95         try {
96             xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
97         } catch (E) {
98             xmlhttp = false;
99         }
100     }
101
102
103     if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
104         xmlhttp = new XMLHttpRequest();
105     }
106     return xmlhttp;
107 }
108
109 var objXMLHttp = new XmlHttp();
110 var objXMLHttp2 = new XmlHttp();
111 var objXMLHttp3 = new XmlHttp();
112 var objXMLHttp4 = new XmlHttp();
113
114 function GetReadyState( ){
115     return this.objXmlHttp.readyState;
116 }
117
118 function GetResponseText( ){
119     return this.objXmlHttp.responseText;
120 }
121
122 function GetUrlContent( strUrl, objOnReadyStateChangeFunction ){
123     this.objXmlHttp.open(this.HttpMethod, strUrl, true);
124     this.objXmlHttp.setRequestHeader( 'Content-Type', 'text/xml;_charset=UTF-8' );
125     if(objOnReadyStateChangeFunction){
126         this.objXmlHttp.onreadystatechange=function(){
127             objOnReadyStateChangeFunction();
128         }
129     }
130     this.objXmlHttp.send( null )
131 }
132
133 //This function is called when we get the data back from the server.
134 function GetResponse( ){
135     if (objXMLHttp.GetReadyState()==4) {
136         // save response in inner html of result object
137         var objMyDiv = document.getElementById( 'LiveVakanzen' );
138         objMyDiv.innerHTML = objXMLHttp.GetResponseText( );
139     }

```

```

140     }
141
142     function GetResponse2( ){
143         if (objXMLHttp2.GetReadyState()==4) {
144             // save response in inner html of result object
145             var objMyDiv2 = document.getElementById( 'MyDiv2' );
146             objMyDiv2.innerHTML = objXMLHttp2.GetResponseText( );
147         }
148     }
149
150     function UpdateDiv( FieldValue , FieldValue2 , FieldValue3 , FieldValue4 ,
151         FieldValue5 , FieldValue6 , FieldValue7 ){
152         if( FieldValue !='' && FieldValue2 !='' && FieldValue3 !='' &&
153             FieldValue4 !='' && FieldValue5 !='' && FieldValue6 !='' && FieldValue7 !='' ){
154             var objMyDiv = document.getElementById( 'LiveVakanzen' );
155             objMyDiv.innerHTML = '<span>Lade Daten</span>';
156             objXMLHttp.GetUrlContent( 'http://' + location.host +
157                 '/ferienhaus-suche/ajax/vakanzabfrage.php?date=' +
158                 FieldValue + '&oref=' + FieldValue2 + '&okat=' + FieldValue3 +
159                 '&days=' + FieldValue4 + '&anbieter=' + FieldValue5 + '&aktion=' +
160                 FieldValue6 + '&sid=' + FieldValue7 , GetResponse );
161         }
162         return;
163     }
164 </script>
165 </head>
166 <body>
167 <!-- *Linkleiste includieren* -->
168 <div NAME="Header" ID="Header">
169 <?php
170 $seite = 'vakanz';
171 $headlinks = $pfad.'/include/suche-headlinks.php';
172 include $headlinks;
173 ?>
174 </div>
175
176 <!-- ***Start Hauptcontainer*** -->
177 <div id="main">
178 <!-- *Einfügen Header (Bild , Region , Name , Objektcode)* -->
179 <?php
180 $bereich = 'vakanz';
181 $inc_header = $pfad.'/ferienhaus-suche/header.php';
182 include $inc_header;
183 ?>
184 <!-- ***Pulldown Monatsauswahl*** -->
185 <!-- ***Container Anreisedatum , Aufenthalt , Buchlink*** -->
186 <div NAME="AuswahlBuchen" ID="AuswahlBuchen" class="vak-reisedaten">
187 <?php
188 $datum_aktuell_buchen = mktime(0, 0, 0, date("m"), date("d"), date("Y"));
189 $datumsfkt->datum_deisoun($aufenthalt_start);
190 $datum_start_buchen = $datumsfkt->datum_unix;
191 ?>
192 <div class="vak_center">Ihre momentane Auswahl:</div>
193 <!-- Solange die Vakanz ungültig ist Link "Buchen" ausblenden -->
194 <table>
195 <tr>
196 <td style="width:150px;"><strong>Anreisedatum: </strong></td>
197 <td><?php echo $aufenthalt_start; ?></td>
198 <tr>
199 <tr>
200 <td style="width:150px;"><strong> Aufenthalt: </strong></td>
201 <td><?php echo $aufenthalt_dauer; ?> Tage</td>
202 <tr>
203 </table>
204 <div class="vak_monatswahl">
205     Bitte Anreisemonat w&#228;hlen

```

```

206         <form>
207         <?php $test = 'hallo'; ?>
208         <select name="L1" size="1" onchange="UpdateDiv( this.value , this.value='<?php
209         .....echo_ $objektreferenz;_?>', this.value='<?php
210         .....echo_ $objektkategorie;_?>',_this.value='<?php
211         .....echo_ $aufenthalt_dauer;_?>',_this.value='<?php
212         .....echo_ $anbieter;_?>',_this.value='kalender ',
213         .....this.value='<?php_echo_ $sessionid;?>'_);" >
214         <?php
215         $year = $datum_aktuell[1];
216         $year2 = $datum_aktuell[1];
217         for ($i = $month_akt; $i < $month_akt+12; $i++) {
218             if ($i < 13) {
219                 $month = $i;
220             }
221             else {
222                 $month = $i-12;
223             }
224
225             if ($i+1 < 13) {
226                 $month2 = $i+1;
227             }
228             else {
229                 $month2 = $i-11;
230             }
231
232             if ($i == 13) {
233                 $year = $year+1;
234             }
235             if ($i+1 == 13) {
236                 $year2 = $year+1;
237             }
238             $optiontext = $datum_array[$month]. ' &nbsp; '. $year. ' &nbsp; - &nbsp; '.
239             $datum_array[$month2]. ' &nbsp; '. $year2;
240             if ($month < 10) {
241                 $month = '0'. $month;
242             }
243             $optionvalue = '15.'. $month. '.'. $year;
244             echo ' <option value="'. $optionvalue. '">'. $optiontext. ' </option>';
245         }
246         ?>
247         <!-- echo ' <textarea cols="20" rows="4" name="textfeld"></textarea>_//-->
248         .....</select>
249         .....</form>
250         .....<!--<span style="color:#FF0000;" _><?php_//echo_ $COOKIE[" Error " ];_?>
251         .....</span><br>_//-->
252         .....<!--***Ende Pulldown Monatsauswahl***_//-->
253
254         .....<?php
255         .....if_($datum_aktuell_buchen+1814400_>=_$datum_start_buchen){
256         .....echo_ ' <span style="color:#FF0000;">Buchung in diesem Zeitraum nicht
257         .....m&#246;glich!</span>';
258         .....}
259         .....else {
260             .....echo ' <a style="text-align: center;"
261             .....href="/ferienhaus-suche/buchung/buchen1.php?sid=' .
262             .....$sessionid. '">Buchen</a>';
263         .....}
264         .....?>
265         .....</div>
266         .....<!-- ***Ende Container Anreisedatum, Aufenthalt, Buchlink***_//-->
267         .....</div>
268
269         .....<!-- ***Start Live Vakanzabfrage***_//-->
270         .....<div NAME="LiveVakanzen" ID="LiveVakanzen" class="vak_vakanzbox">
271         .....<span>Vakanzen</span>

```

```

272     <?php
273     switch ($anbieter) {
274         case 1:
275             include $interhome.vakanz_live;
276             break;
277         case 2:
278             include $wolters.vakanz_live;
279             break;
280         case 99:
281             include $onfh.vakanz_live;
282             break;
283     }
284     ?>
285     <!-- ***Ende Live Vakanzabfrage*** -->
286 </div>
287
288 <div class="vak_farbschema">
289     <span class="vak_tabelle_tage_frei" >&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>Objekt frei&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</div>
290     <span class="vak_tabelle_tage_belegt" >&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>
291     Objekt belegt&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</div>
292     <span class="vak_tabelle_tage_anfrage" >&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>Auf Anfrage
293 </div>
294 <!-- ***Start Zusatzleistungen*** -->
295 <div NAME="Zusatzleistungen" ID="Zusatzleistungen" class="vak_zubs">
296     Zusatzleistungen:<br>
297     <?php
298     for ($zubs_count = 0 ; $zubs_count < $anzahl_zubs; $zubs_count++) {
299         echo '<strong>'.utf8_encode($zubs[$zubs_count]['Leistung']).'</strong><br>';
300         echo ' '.utf8_encode($zubs[$zubs_count]['Serviceart']).' ';
301         echo $zubs[$zubs_count]['Preis'];
302         echo $zubs[$zubs_count]['Waehrung'].', ' . ' ';
303         echo utf8_encode($zubs[$zubs_count]['Berechnung']).' ';
304         echo '<br><br>';
305     }
306     ?>
307     <!-- ***Ende Zusatzleistungen*** -->
308 </div>
309 <!-- ***Ende Hauptcontainer*** -->
310 </div>
311 </body>
312 </html>

```

G.6. Vakanzausgabe allgemein

Die Ausgabe der Vakanzen ist bei allen Anbietern in weiten Teilen identisch. Während die Daten für Interhome und private Objekte aus der Datenbank bezogen werden, erfolgt bei Wolters Objekten eine Abfrage des Webservice und Konvertierung der Ergebnisse. Deshalb im folgenden das Script der Wolters Vakanzausgabe.

Listing G.6: Quelltext Überprüfungsroutine für Eingaben im Frontend

```

1 <?php
2
3 /**
4  * Wolters Verfügbarkeitsabfrage
5  * Script holt sich die Verfügbarkeitsdaten
6  * direkt und aktuell vom Wolters Server
7  * und berücksichtigt sie in der Ausgabe
8  * des Kalenders
9  */
10
11 $java = 1;
12 if($_REQUEST['nj'] == 1) {

```



```

13  $java = 0;
14  }
15
16  $datum_monat1 = explode(".", $aufenthalt_start);
17  $jahr = $datum_monat1[2];
18  $monat = $datum_monat1[1];
19
20  /* *Ermittlung Anz. Tage des aktuellen / angegebenen Monats* */
21  $datumsfkt->days_act_month($datum_monat1);
22  $days_mon_now = $datumsfkt->daysamon;
23  $lastdaymon1 = $days_mon_now; //temporär
24
25  /* *Ermittlung Anz. Tage des Folgemonats* */
26  $datumsfkt->days_next_month($datum_monat1);
27  $days_mon_next = $datumsfkt->daysnmon;
28  $lastdaymon2 = $days_mon_next; //temporär
29
30  /* *Start und Ende des gewählten Aufenthaltsmonats ermitteln, im DATE Format* */
31  $datumsfkt->startendactmonth($datum_monat1, $days_mon_now);
32  $datum_monat1_start = $datumsfkt->startdmon;
33  $datum_monat1_ende = $datumsfkt->enddmon;
34
35  /* *Folgemonat ermitteln* */
36  $datumsfkt->thistonextmonth($datum_monat1[1]);
37  $datum_monat_vor = $datumsfkt->nextmonth;
38
39  /* *Start und Endtag Folgemonat ermitteln* */
40  $datumsfkt->startendnexmonth($datum_monat1, $days_mon_next, $datum_monat_vor);
41  $startdatum_next = $datumsfkt->startdnmon;
42  $enddatum_next = $datumsfkt->enddnmon;
43
44  $datum_monat2_start = $startdatum_next; //Temporär
45  $datum_monat2_ende = $enddatum_next; //Temporär
46
47  /* *1. Monat an die Wolters Abfrage geben* */
48  $startdatum = $datum_monat1_start;
49  $enddatum = $datum_monat1_ende;
50  $objektcodecat = $objektreferenz.'-'. $objektkategorie;
51  //Fragt Monatsmitte ab, Wolters Response deckt dann fast kompletten Monat ab
52  $datum_xml = $datum_monat1[2].'-'. $datum_monat1[1].'-15';
53  $abfrageurl = "http://www.tui-ferienhaus.de/xmlvakanz.php?obj_cat="
54  $objektcodecat&date=$datum_xml&dummya=53088";
55  $fetchcontent = fopen ($abfrageurl, "r");
56  while (!feof($fetchcontent)) {
57      $savecontent = fread($fetchcontent, 4096);
58  }
59  fclose($fetchcontent);
60
61  $xml_vakanz = new SimpleXMLElement($savecontent);
62  $i_wovakanz = 0;
63  foreach($xml_vakanz->date as $vakanzdate) {
64      $wolters_vakanzarray[$i_wovakanz] = array((string)$vakanzdate,
65      (string)$vakanzdate['flag']);
66      $i_wovakanz++;
67  }
68  $vakanzwochen = count($wolters_vakanzarray);
69  $firstrun = 1;
70  for ($i_wovakanz = 0; $i_wovakanz < $vakanzwochen; $i_wovakanz++) {
71      $split_wodate = explode("-", $wolters_vakanzarray[$i_wovakanz][0]);
72
73      /* *Monat vor aktuellem Monat* */
74      if ($split_wodate[1] == $datum_monat1[1]-1) {
75          /**Setzen Zeiger für alle Tage vor dem 1. Samstag des Buchungs / Anfragemonats**/
76          if ($wolters_vakanzarray[$i_wovakanz][1] == 1) {
77              $status_monthbefore = 1;
78          }

```

```

79     else { $status_monthbefore = 0; }
80 }
81 else {
82     $status_monthbefore = 'NULL';
83 }
84 /* — Generierung eines Interhome ähnlichen Vakanzarray mit den Buchstaben Y und N
85    dadurch kann der Rest des scriptes direkt vom Interhome Kalenderscript genommen
86    werden — */
87
88 /* *Aktueller Buchungs / Anfragemonat* */
89 if ($split_wodate[1] == $datum_month1[1]) {
90     /* Array beginnend ab Null, +1 für das korrekte Datum addieren */
91     $arraystart = $split_wodate[2]-1;
92     $arrayende = $split_wodate[2]+5;
93     if ($firstrun == 1) {
94         for ($i_array = 0; $i_array <= $arraystart; $i_array++) {
95             if ($status_monthbefore == 0) {
96                 $wo_char_vakanzarray[$i_array] = 'Y';
97             }
98             if ($status_monthbefore == 1) {
99                 $wo_char_vakanzarray[$i_array] = 'N';
100             }
101         }
102         $firstrun = 0;
103     }
104     for ($i_array = $arraystart; $i_array <= $arrayende; $i_array++) {
105
106         if ($wolters_vakanzarray[$i_wovakanz][1] == 0) {
107             $wo_char_vakanzarray[$i_array] = 'Y';
108         }
109         if ($wolters_vakanzarray[$i_wovakanz][1] == 1) {
110             $wo_char_vakanzarray[$i_array] = 'N';
111         }
112     }
113 }
114 if ($split_wodate[1] == $datum_month1[1]+1) {
115     //echo 'Folgend';
116 }
117 }
118
119 /* *fertiges Vakanzarray* */
120 $wochenarray = $wo_char_vakanzarray;
121
122 /* —Aufsetzen des Interhome Vakanz-Tabellenscriptes— */
123 /* *Tabellenscript Tabelle 1* */
124 $datumsfkt->tabelle_monatskopf($datum_month1);
125 $tabellenkopf_mon = $datumsfkt->tblkopfmon;
126 ?>
127 <!-- *Tabellenkopf mit Monatsangabe* /-->
128 <div class="vak_montable1">
129     <table class="vak_montablehead">
130         <tr><th><?php echo $tabellenkopf_mon; ?></th></tr>
131     </table>
132 <?php
133     /* *Korrekte Positionierung des Starttages durch Vergleich $tageszaehler mit
134        Wochentag 1. Tag des entspr. Monats. Arraywert als Startpositon* */
135     $tageszaehler = array('Monday' => 1, 'Tuesday' => 2, 'Wednesday' => 3, 'Thursday'
136     => 4, 'Friday' => 5, 'Saturday' => 6, 'Sunday' => 7);
137     $monatsbeginn = strtotime($datum_month1[2].'-'. $datum_month1[1].'-01');
138     /* *Ermittlung Name des Tages* */
139     $wochentag_start = strftime("%A", $monatsbeginn);
140     /* *Ermittlung Startposition in Tabelle* */
141     $counter_start = $tageszaehler[$wochentag_start];
142     ?>
143     <!-- *Tabelle 1* /-->
144     <table class="vak_montabledays">

```

```

145 <!-- *1. Zeile Wochentage, danach Nummernausgabe* //-->
146 <tr>
147 <td class="vak_td">Mo</td><td class="vak_td">Di</td><td
148 class="vak_td">Mi</td><td class="vak_td">Do</td><td class="vak_td">Fr</td><td
149 class="vak_td">Sa</td><td class="vak_td">So</td>
150 </tr>
151 <!-- ***Start Bereich Tagesausgabe*** //-->
152 <tr>
153 <?php
154 $k = 1;
155 $tag = $k-1; // Da Tage in einem Array beginnend mit Null
156 for ($i = 1; $i <= 42; $i++) { //6 Wochen a 7 Tage
157     if ($i == 8 OR $i == 15 OR $i == 22 OR $i == 29 OR $i == 36) { // Montage =
158         neue Zeile
159         echo '<tr>';
160     }
161     if ($i >= $counter_start) {
162         if ($k <= $lastdaymon1) {
163             if ($k < 10) {
164                 $tag2 = '0'.$k;
165             }
166             else {
167                 $tag2 = $k;
168             }
169             if ($wochenarray[$tag] == 'Y' AND $i != 6 AND $i != 13 AND $i != 20 AND $i
170 != 27 AND $i != 34 AND $i != 41 ) {
171                 echo '<td class="vak_tabelle_tage_frei">'.$k.'</td>';
172             }
173             if ($java != 0) {
174                 if ($wochenarray[$tag] == 'Y' AND $i == 6) {
175                     echo '<td class="vak_tabelle_tage_frei"><a href="#"
176 ..... onclick="UpdateDiv(this.value=\'\'.'$jahr.'-'.$monat.'-'.$
177 ..... $tag2.'\'',this.value=\'\'.'$objektreferenz.'\'',
178 ..... this.value=\'\'.'$objektkategorie.'\'',this.value=\'\'.'
179 ..... $aufenthalt_dauer.'\'',this.value=\'\'.'$anbieter.'\'',
180 ..... this.value=\'preis\'',this.value=\'\'.'$sessionid.'\'');">'.$k.'</a></td>';
181                 }
182                 if ($wochenarray[$tag] == 'Y' AND $i == 13) {
183                     echo '<td class="vak_tabelle_tage_frei"><a href="#" onclick=
184 ..... "UpdateDiv(this.value=\'\'.'$jahr.'-'.$monat.'-'.$tag2.'\'',
185 ..... this.value=\'\'.'$objektreferenz.'\'',this.value=\'\'.'$objektkategorie.'\'',
186 ..... this.value=\'\'.'$aufenthalt_dauer.'\'',this.value=\'\'.'$anbieter.'\'',
187 ..... this.value=\'preis\'',this.value=\'\'.'$sessionid.'\'');">'.$k.'
188 ..... </a></td>';
189                 }
190                 if ($wochenarray[$tag] == 'Y' AND $i == 20) {
191                     echo '<td class="vak_tabelle_tage_frei"><a href="#"
192 ..... onclick="UpdateDiv(this.value=\'\'.'$jahr.'-'.$
193 ..... $monat.'-'.$tag2.'\'',this.value=\'\'.'$objektreferenz.'
194 ..... \'\'',this.value=\'\'.'$objektkategorie.'\'',this.value=\'\'.'
195 ..... $aufenthalt_dauer.'\'',this.value=\'\'.'$anbieter.'\'',
196 ..... this.value=\'preis\'',this.value=\'\'.'
197 ..... $sessionid.'\'');">'.$k.'</a></td>';
198                 }
199                 if ($wochenarray[$tag] == 'Y' AND $i == 27) {
200                     echo '<td class="vak_tabelle_tage_frei"><a href="#"
201 ..... onclick="UpdateDiv(this.value=\'\'.'$jahr.'-'.$monat.'
202 ..... '-'.'$tag2.'\'',this.value=\'\'.'$objektreferenz.'
203 ..... \'\'',this.value=\'\'.'$objektkategorie.'\'',
204 ..... this.value=\'\'.'$aufenthalt_dauer.'\'',
205 ..... this.value=\'\'.'$anbieter.'\'',this.value=\'preis\'',
206 ..... this.value=\'\'.'$sessionid.'\'');">'.$k.'</a></td>';
207                 }
208                 if ($wochenarray[$tag] == 'Y' AND $i == 34) {
209                     echo '<td class="vak_tabelle_tage_frei"><a href="#"
210 ..... onclick="UpdateDiv(this.value=\'\'.'$jahr.'-'.$monat.'-'.$tag2.'\'',

```

```

211 .....this.value=''. $objektreferenz. '\', this.value=''.
212 ..... $objektkategorie. '\', this.value=''. $aufenthalt_dauer. '\',
213 .....this.value=''. $anbieter. '\', this.value='preis\'',
214 .....this.value=''. $sessionid. '\');">'. $k. '</a></td>';
215 .....}
216 .....}
217 .....else {
218 .....    if ($wochenarray[$tag] == 'Y' AND $i == 6) {
219 .....        echo '<td_class="vak_tabelle_tage_frei"><a
220 .....        href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. '.$
221 .....        $jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
222 .....    }
223 .....    if ($wochenarray[$tag] == 'Y' AND $i == 13) {
224 .....        echo '<td_class="vak_tabelle_tage_frei"><a
225 .....        href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. '.$
226 .....        $jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
227 .....    }
228 .....    if ($wochenarray[$tag] == 'Y' AND $i == 20) {
229 .....        echo '<td_class="vak_tabelle_tage_frei"><a
230 .....        href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. '.$
231 .....        $jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
232 .....    }
233 .....    if ($wochenarray[$tag] == 'Y' AND $i == 27) {
234 .....        echo '<td_class="vak_tabelle_tage_frei"><a
235 .....        href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. '.$
236 .....        $jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
237 .....    }
238 .....    if ($wochenarray[$tag] == 'Y' AND $i == 34) {
239 .....        echo '<td_class="vak_tabelle_tage_frei"><a
240 .....        href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. '.$
241 .....        $jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
242 .....    }
243 .....}
244 .....if ($wochenarray[$tag] == 'N') {
245 .....    echo '<td_class="vak_tabelle_tage_belegt">'. $k. '</td>';
246 .....}
247 .....if ($wochenarray[$tag] == 'Q' OR $wochenarray[$tag] == 'K') {
248 .....    echo '<td_class="vak_tabelle_tage_anfrage">'. $k. '</td>';
249 .....}
250 .....if ($wochenarray[$tag] == '') {
251 .....    echo '<td<font_style="color:#000000;">'. $k. '</font></td>';
252 .....}
253 .....$k++; $tag++;
254 .....}
255 .....else { echo '<td>&#160;</td>'; }
256 .....}
257 .....else { echo '<td>&#160;</td>'; }
258 .....if ($i == 7 OR $i == 14 OR $i == 21 OR $i == 28 OR $i == 35) { // Sonntage =
259 .....Zeilenende
260 .....    echo '</tr>';
261 .....}
262 .....
263 .....}
264 .....?>
265 .....<!-- ***Ende Bereich Tagesausgabe*** //—>
266 .....</tr>
267 .....</table>
268 .....<?php
269 .....$dlmonzur = $datum_month1[1]-1;
270 .....if ($dlmonzur < 10) {
271 .....    $dlmonzur = '0'. $dlmonzur;
272 .....}
273 .....?>
274 .....<!-- Monatsausgabe1 Ende //—>
275 .....</div>
276 .....<?php

```

```

277 /* ——— 2. Monat an die Wolters Abfrage geben ——— */
278 $startdatum = $datum_monat2_start;
279 $enddatum = $datum_monat2_ende;
280 $objektcodecat = $objektreferenz.'-'. $objektkategorie;
281
282 $datum_month2 = $datum_month1[1]+1;
283 if ($datum_month2 < 10) {
284     $datum_month2 = '0'. $datum_month2;
285 }
286
287 $datum_xml = $datum_month1[2].'-'. $datum_month2.'-15'; //Fragt Monatsmitte ab,
288 Wolters Response deckt dann fast kompletten Monat ab
289 $abfrageurl = "http://www.tui-ferienhaus.de/xmlvakanz.php?
290 obj_cat=$objektcodecat&date=$datum_xml&dummya=53088";
291 $fetchcontent = fopen ($abfrageurl,"r");
292 while (!feof($fetchcontent)) {
293     $savecontent = fread($fetchcontent , 4096);
294 }
295 fclose($fetchcontent);
296
297 $xml_vakanz = new SimpleXMLElement($savecontent);
298 $i_wovakanz = 0;
299 foreach($xml_vakanz->date as $vakanzdate) {
300     $wolters_vakanzarray[$i_wovakanz] = array((string)$vakanzdate ,
301         (string)$vakanzdate['flag']);
302     $i_wovakanz++;
303 }
304
305 $vakanzwochen = count($wolters_vakanzarray);
306 $firstrun = 1;
307 for ($i_wovakanz = 0; $i_wovakanz < $vakanzwochen; $i_wovakanz++) {
308     $split_wodate = explode("-", $wolters_vakanzarray[$i_wovakanz][0]);
309
310     if ($split_wodate[1] == $datum_month1[1]) {
311         // Setzen Zeiger für alle Tage vor dem 1. Samstag des Buchungs / Anfragemonats
312         if ($wolters_vakanzarray[$i_wovakanz][1] == 1) {
313             $status_monthbefore = 1;
314         }
315         else { $status_monthbefore = 0; }
316         //echo 'Mbefore';
317     }
318     //Aktueller Buchungs / Anfragemonat
319     if ($split_wodate[1] == $datum_month1[1]+1) {
320         //Array beginnend ab Null, +1 für das korrekte Datum addieren
321         $arraystart = $split_wodate[2]-1;
322         $arrayende = $split_wodate[2]+5;
323         if ($firstrun == 1) {
324             for ($i_array = 0; $i_array <= $arraystart; $i_array++) {
325                 if ($status_monthbefore == 0) {
326                     $wo_char_vakanzarray[$i_array] = 'Y';
327                 }
328                 if ($status_monthbefore == 1) {
329                     $wo_char_vakanzarray[$i_array] = 'N';
330                 }
331             }
332             $firstrun = 0;
333         }
334         for ($i_array = $arraystart; $i_array <= $arrayende; $i_array++) {
335
336             if ($wolters_vakanzarray[$i_wovakanz][1] == 0) {
337                 $wo_char_vakanzarray[$i_array] = 'Y';
338             }
339             if ($wolters_vakanzarray[$i_wovakanz][1] == 1) {
340                 $wo_char_vakanzarray[$i_array] = 'N';
341             }
342         }

```

```

343     }
344 }
345 if ($split_wodate[1] == $datum_month1[1]+2) {
346 }
347 }
348 $wochenarray = $wo_char_vakanzarray;
349 ?>
350
351 <!-- *Tabellenscript Tabelle 2* //-->
352 <!-- ***Monatsausgabe2 Anfang*** //-->
353 <div class="vak_montable2">
354     <?php
355         $datum_month1vor[1] = $datum_month1[1]+1;
356         $datum_month1vor[2] = $datum_month1[2];
357         $datumsfkt->tabelle_monatskopf($datum_month1vor);
358         $tabellenkopf_mon = $datumsfkt->tblkopfmon;
359     ?>
360     <!-- *Monatstabellenkopf* //-->
361     <table class="vak_montablehead">
362         <tr><th><?php echo $tabellenkopf_mon; ?></th></tr>
363     </table>
364     <?php
365     if ($datum_month1[1]+1 == 13) {
366         $datum_month1vor = $datum_month1[2]+1;
367         $monatsbeginn = strtotime($datum_month1vor.'-01-01');
368     }
369     else {
370         $monatvor = $datum_month1[1]+1;
371         if ($monatvor < 10) {
372             $monatvor = '0'.$monatvor;
373         }
374
375         $monatsbeginn = strtotime($datum_month1[2].'-'.$monatvor.'-01');
376     }
377     $wochentag_start = strftime("%A", $monatsbeginn);
378     $counter_start = $tageszaehler[$wochentag_start];
379     ?>
380     <table class="vak_montabledays">
381         <tr>
382             <td class="vak_td">Mo</td><td class="vak_td">Di</td><td
383                 class="vak_td">Mi</td><td class="vak_td">Do</td><td class="vak_td">Fr</td><td
384                 class="vak_td">Sa</td><td class="vak_td">So</td>
385         </tr>
386         <!-- ***Start Bereich Tagesausgabe*** //-->
387         <tr>
388             <?php
389             /* *einen Monat vor setzen* */
390             if ($monat < 9) {
391                 $monat = $monat+1;
392                 $monat = '0'.$monat;
393             }
394             else {
395                 $monat = $monat+1;
396             }
397             $k = 1;
398             $tag = $k-1;
399             for ($i = 1; $i <= 42; $i++) {
400                 if ($i == 8 OR $i == 15 OR $i == 22 OR $i == 29 OR $i == 36) {
401                     /* <tr> -> Start, Woche */
402                     echo '<tr>';
403                 }
404                 if ($i >= $counter_start) {
405                     if ($k <= $lastdaymon2) {
406                         //verhindern von einstelligen Tageszahlen
407                         if ($k < 10) {
408                             $tag2 = '0'.$k;

```

```

409     }
410     else {
411         $tag2 = $k;
412     }
413     if ($wochenarray[$tag] == 'Y' AND $i != 6 AND $i != 13 AND $i != 20 AND $i
414     != 27 AND $i != 34 AND $i != 41 ) {
415         echo '<td_class="vak_tabelle_tage_frei">'. $k. '</td>';
416     }
417     if ($java != 0) {
418         if ($wochenarray[$tag] == 'Y' AND $i == 6) {
419             echo '<td_class="vak_tabelle_tage_frei"><a_href="#"
420             onclick="UpdateDiv(this.value=\'\'.$jahr.'-'. $monat.'-'. $tag2.
421             '\',this.value=\'\'.$objektreferenz.'\'',this.value=\'\'
422             $objektkategorie.'\'',this.value=\'\'.$aufenthalt_dauer.'\'',
423             this.value=\'\'.$anbieter.'\'',this.value=\'preis\'',
424             this.value=\'\'.$sessionid.'\'');">'. $k. '</a></td>';
425         }
426         if ($wochenarray[$tag] == 'Y' AND $i == 13) {
427             echo '<td_class="vak_tabelle_tage_frei"><a_href="#"
428             onclick="UpdateDiv(this.value=\'\'.$jahr.'-'. $monat.'-'. $tag2.'\'',
429             this.value=\'\'.$objektreferenz.'\'',this.value=\'\'
430             $objektkategorie.'\'',this.value=\'\'.$aufenthalt_dauer.'\'',
431             this.value=\'\'.$anbieter.'\'',this.value=\'preis\'',
432             this.value=\'\'.$sessionid.'\'');">'. $k. '</a></td>';
433         }
434         if ($wochenarray[$tag] == 'Y' AND $i == 20) {
435             echo '<td_class="vak_tabelle_tage_frei"><a_href="#"
436             onclick="UpdateDiv(this.value=\'\'.$jahr.'-'. $monat.'-'. $tag2.
437             '\',this.value=\'\'.$objektreferenz.'\'',
438             this.value=\'\'.$objektkategorie.'\'',
439             this.value=\'\'.$aufenthalt_dauer.'\'',this.value=\'\'.$anbieter.'\'',
440             this.value=\'preis\'',this.value=\'\'.$sessionid.'\'');">'.
441             $k. '</a></td>';
442         }
443         if ($wochenarray[$tag] == 'Y' AND $i == 27) {
444             echo '<td_class="vak_tabelle_tage_frei"><a_href="#"
445             onclick="UpdateDiv(this.value=\'\'.$jahr.'-'. $monat.'-'. $tag2.
446             '\',this.value=\'\'.$objektreferenz.'\'',
447             this.value=\'\'.$objektkategorie.'\'',
448             this.value=\'\'.$aufenthalt_dauer.'\'',this.value=\'\'.$anbieter.'\'',
449             this.value=\'preis\'',this.value=\'\'.$sessionid.'\'');">'.
450             $k. '</a></td>';
451         }
452         if ($wochenarray[$tag] == 'Y' AND $i == 34) {
453             echo '<td_class="vak_tabelle_tage_frei"><a_href="#"
454             onclick="UpdateDiv(this.value=\'\'.$jahr.'-'. $monat.'-'.
455             $tag2.'\'',this.value=\'\'.$objektreferenz.'\'',
456             this.value=\'\'.$objektkategorie.'\'',
457             this.value=\'\'.$aufenthalt_dauer.'\'',this.value=\'\'.$anbieter.'\'',
458             this.value=\'preis\'',this.value=\'\'.$sessionid.'\'');">'.
459             $k. '</a></td>';
460         }
461     }
462     else {
463         if ($wochenarray[$tag] == 'Y' AND $i == 6) {
464             echo '<td_class="vak_tabelle_tage_frei"><a
465             href="vakanzen-nj.php?beginn=\'\'.$tag2.'\''. $monat.'\''.
466             $jahr.'&nj=1&sid=\'\'.$sessionid.'">'. $k. '</a></td>';
467         }
468         if ($wochenarray[$tag] == 'Y' AND $i == 13) {
469             echo '<td_class="vak_tabelle_tage_frei"><a
470             href="vakanzen-nj.php?beginn=\'\'.$tag2.'\''. $monat.'\''.
471             $jahr.'&nj=1&sid=\'\'.$sessionid.'">'. $k. '</a></td>';
472         }
473         if ($wochenarray[$tag] == 'Y' AND $i == 20) {
474             echo '<td_class="vak_tabelle_tage_frei"><a

```

```

475 .....href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. ' .
476 .....$jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
477 .....}
478 .....if ($wochenarray[$tag] == 'Y' AND $i == 27) {
479 .....echo '<td class="vak_tabelle_tage_frei"><a
480 .....href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. ' .
481 .....$jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
482 .....}
483 .....if ($wochenarray[$tag] == 'Y' AND $i == 34) {
484 .....echo '<td class="vak_tabelle_tage_frei"><a
485 .....href="vakanzen-nj.php?beginn='.$tag2.'. '.$monat.'. ' .
486 .....$jahr. '&nj=1&sid='.$sessionid.'">'. $k. '</a></td>';
487 .....}
488 .....}
489 .....if ($wochenarray[$tag] == 'N') {
490 .....echo '<td class="vak_tabelle_tage_belegt">'. $k. '</td>';
491 .....}
492 .....if ($wochenarray[$tag] == 'Q' OR $wochenarray[$tag] == 'K') {
493 .....echo '<td class="vak_tabelle_tage_anfrage">'. $k. '</td>';
494 .....}
495 .....if ($wochenarray[$tag] == '') {
496 .....echo '<td style="color:#000000;">'. $k. '</td>';
497 .....}
498 .....$k++; $tag++;
499 .....}
500 .....else { echo '<td>&#160;</td>'; }
501 .....}
502 .....else { echo '<td>&#160;</td>'; }
503 .....if ($i == 7 OR $i == 14 OR $i == 21 OR $i == 28 OR $i == 35) {
504 ...../* </tr> -> Ende, Woche */
505 .....echo '</tr>';
506 .....}
507 .....}
508 .....?>
509 .....<!-- ***Ende Bereich Tagesausgabe*** //—>
510 .....</tr>
511 .....</table>
512 .....<?php
513 .....$dlmonvor = $datum_monat[1]+1;
514 .....if ($dlmonvor < 10) {
515 .....$dlmonvor = '0'.$dlmonvor;
516 .....}
517 .....?>
518 .....<!-- ***Monatsausgabe2 Ende*** //—>
519 .....</div>

```

G.7. Preisausgabe Interhome Objekte

Wie bereits in der Arbeit erwähnt ist das Script mit dem für Wolters Objekte vom Prinzip her gleich. Zudem entspricht es nicht mehr dem aktuellen Stand. Da jedoch noch keine neue Variante vorliegt kann sie an dieser Stelle nicht präsentiert werden.

G.8. Preisausgabe Wolters Objekte

Listing G.7: Quelltext Preisausgabe Wolters

```

1 <?php
2

```



```

3 /**
4  * Live Preisabfrage für Wolters Vakanzen
5  * Holt sich tagesaktuelle Preise direkt über
6  * die Webservices von Wolters und gibt sie in
7  * einer Vorschau von 1 bis 4 Wochen wieder.
8 */
9
10 $java = 1;
11 if ($.GET['nj'] == 1) {
12     $java = 0;
13 }
14 $objrefkat = $objektreferenz.'-'. $objektkategorie;
15 $aufenthalt_datum_array = explode("-", $aufenthalt_start);
16 /* Alternative, wenn format dd.mm.yyyy */
17 if ($aufenthalt_datum_array[2] == '') {
18     $aufenthalt_datum_array1 = explode(".", $aufenthalt_start);
19     $aufenthalt_datum_array[0] = $aufenthalt_datum_array1[2];
20     $aufenthalt_datum_array[1] = $aufenthalt_datum_array1[1];
21     $aufenthalt_datum_array[2] = $aufenthalt_datum_array1[0];
22 }
23
24 $startdatum_unix = mktime(0,0,0,$aufenthalt_datum_array[1],
25 $aufenthalt_datum_array[2],$aufenthalt_datum_array[0]);
26 $woche1_ende_unix = $startdatum_unix+604800;
27 $woche2_ende_unix = $startdatum_unix+1209600;
28 $woche3_ende_unix = $startdatum_unix+1814400;
29 $woche4_ende_unix = $startdatum_unix+2419200;
30
31 $woche1_start = strftime("%d.%m.%Y", $startdatum_unix);
32 $woche1_ende = strftime("%d.%m.%Y", $woche1_ende_unix);
33 $woche2_ende = strftime("%d.%m.%Y", $woche2_ende_unix);
34 $woche3_ende = strftime("%d.%m.%Y", $woche3_ende_unix);
35 $woche4_ende = strftime("%d.%m.%Y", $woche4_ende_unix);
36
37 $woche1_start_iso = strftime("%Y-%m-%d", $startdatum_unix);
38 $woche1_ende_iso = strftime("%Y-%m-%d", $woche1_ende_unix);
39 $woche2_ende_iso = strftime("%Y-%m-%d", $woche2_ende_unix);
40 $woche3_ende_iso = strftime("%Y-%m-%d", $woche3_ende_unix);
41 $woche4_ende_iso = strftime("%Y-%m-%d", $woche4_ende_unix);
42 ?>
43 <table>
44     <tr>
45         <td width="200px">
46             Aufenthalt von bis
47         </td>
48         <td width="100px">
49             Preis
50         </td>
51         <td width="100px">
52             Dauer
53         </td>
54     </tr>
55     <!-- Woche 1 /-->
56     <?php
57     $abfrageurl = "http://www.tui-ferienhaus.de/agent-xml-export.php?
58     _price&datum.kz=$woche1_start_iso&dauer.kz=1&obj_cat[]=$objrefkat";
59     $fetchcontent = fopen ($abfrageurl,"r");
60     while (!feof($fetchcontent)) {
61         $savecontent = fread($fetchcontent, 4096);
62     }
63     fclose($fetchcontent);
64     $xml_prices = new SimpleXMLElement($savecontent);
65     foreach($xml_prices->Kontingent as $preise) {
66         if ((int)$preise['BelegtAnzahl'] == 0) {
67             $preis = $preise->preis_1W.'EUR';
68             if ($preis != 0) {

```

```

69     ?>
70     <tr class="vak_preistr1">
71         <td width="300px">
72             <?php
73                 if ($java == 1) {
74                     ?>
75                     <a href="vakanzen.php?oref=<?php_echo_$objektreferenz;_?>&okat=<?php
76 .....echo_$objektkategorie;_?>&date=<?php_echo_$woche1_start;
77 .....?>&days=7&anbieter=<?php_echo_$anbieter;_?>&p=<?php_echo
78 .....$preis;?>&view=1&sid=<?php_echo_$sessionid;?>"><?php echo
79 .....$woche1_start.'_bis_'. $woche1_ende; ?></a>
80                 <?php
81                 }
82                 else {
83                     ?>
84                     <a href="vakanzen-nj.php?oref=<?php_echo_$objektreferenz;
85 .....?>&okat=<?php_echo_$objektkategorie;_?>&date=<?php_echo_$woche1_start;
86 .....?>&days=7&anbieter=<?php_echo_$anbieter;_?>&p=<?php_echo
87 .....$preis;?>&view=1&nj=1&sid=<?php_echo_$sessionid;?>"><?php echo
88 .....$woche1_start.'_bis_'. $woche1_ende; ?></a>
89                 <?php
90                 }
91                 ?>
92             </td>
93             <td width="100px">
94                 <?php
95                     echo $preise->preis_1W.'&nbsp;EUR';
96                 ?>
97             </td>
98             <td width="100px">
99                 7 Tage
100            </td>
101        </tr>
102    <?php
103    }
104    }
105    }
106    ?>
107
108    <!-- *Woche 2* /-->
109    <?php
110        $abfrageurl = "http://www.tui-ferienhaus.de/agent_xml_export.php?
111    _price&datum_kz=$woche1_start_iso&dauer_kz=2&obj_cat[]=$objrefkat";
112        $fetchcontent = fopen($abfrageurl,"r");
113        while (!feof($fetchcontent)) {
114            $savecontent = fread($fetchcontent, 4096);
115        }
116        fclose($fetchcontent);
117        $xml_prices = new SimpleXMLElement($savecontent);
118        foreach($xml_prices->Kontingent as $preise) {
119            if ((int)$preise['BelegtAnzahl'] == 0) {
120                $preis = $preise->preis_2W.'EUR';
121                if ($preis != 0) {
122                    ?>
123                    <tr class="vak_preistr2">
124                        <td width="300px">
125                            <?php
126                                if ($java == 1) {
127                                    ?>
128                                    <a href="vakanzen.php?oref=<?php_echo_$objektreferenz;_?>&okat=<?php
129 .....echo_$objektkategorie;_?>&date=<?php_echo_$woche1_start;
130 .....?>&days=14&anbieter=<?php_echo_$anbieter;_?>&p=<?php_echo
131 .....$preis;?>&view=1&sid=<?php_echo_$sessionid;?>"><?php echo
132 .....$woche1_start.'_bis_'. $woche2_ende; ?></a>
133                                <?php
134                            }

```

```

135         else {
136             ?>
137             <a href="vakanzen-nj.php?oref=<?php echo $objektreferenz;
138 .....>?&okat=<?php echo $objektkategorie; ?>&date=<?php echo
139 .....$woche1_start; ?>&days=14&anbieter=<?php echo $anbieter; ?>&p=<?php
140 .....echo $preis; ?>&view=1&nj=1&sid=<?php echo $sessionid; ?>"><?php echo
141 .....$woche1_start.' _bis_ '. $woche2_ende; ?></a>
142         <?php
143         }
144         ?>
145     </td>
146     <td width="100px">
147         <?php
148         echo $preise->preis_2W.' &nbsp; EUR';
149         ?>
150     </td>
151     <td width="100px">
152         14 Tage
153     </td>
154 </tr>
155 <?php
156 }
157 }
158 }
159 ?>
160
161 <!-- *Woche 3* //-->
162 <?php
163 $abfrageurl = "http://www.tui-ferienhaus.de/agent_xml_export.php?
164 ..price&datum_kz=$woche1_start_iso&dauer_kz=3&obj_cat[]=$objrefkat";
165 $fetchcontent = fopen ($abfrageurl,"r");
166 while (!feof($fetchcontent)) {
167     $savecontent = fread($fetchcontent, 4096);
168 }
169 fclose ($fetchcontent);
170 $xml_prices = new SimpleXMLElement($savecontent);
171 foreach ($xml_prices->Kontingent as $preise) {
172     if ((int)$preise['BelegtAnzahl'] == 0) {
173         $preis = $preise->preis_3W.' EUR';
174         if ($preis != 0) {
175             ?>
176             <tr class="vak_preistr1">
177                 <td width="300px">
178                     <?php
179                     if ($java == 1) {
180                         ?>
181                         <a href="vakanzen.php?oref=<?php echo $objektreferenz; ?>&okat=<?php
182 .....echo $objektkategorie; ?>&date=<?php echo $woche1_start;
183 .....>?&days=21&anbieter=<?php echo $anbieter; ?>&p=<?php echo
184 .....$preis; ?>&view=1&sid=<?php echo $sessionid; ?>"><?php echo
185 .....$woche1_start.' _bis_ '. $woche3_ende; ?></a>
186                     <?php
187                     }
188                     else {
189                         ?>
190                         <a href="vakanzen-nj.php?oref=<?php echo $objektreferenz;
191 .....>?&okat=<?php echo $objektkategorie; ?>&date=<?php echo $woche1_start;
192 .....>?&days=21&anbieter=<?php echo $anbieter; ?>&p=<?php echo
193 .....$preis; ?>&view=1&nj=1&sid=<?php echo $sessionid; ?>"><?php echo
194 .....$woche1_start.' _bis_ '. $woche3_ende; ?></a>
195                     <?php
196                     }
197                     ?>
198                 </td>
199                 <td width="100px">
200                     <?php

```

```

201         echo $preise->preis_3W . '&nbsp;EUR' ;
202     ?>
203 </td>
204 <td width="100px">
205     21 Tage
206 </td>
207 </tr>
208 <?php
209     }
210 }
211 }
212 ?>
213
214 <!-- *Woche* 4 /-->
215 <?php
216 $abfrageurl = "http://www.tui-ferienhaus.de/agent_xml-export.php?
217 price&datum_kz=$woche1_start_iso&dauer_kz=4&obj_cat[]=$objrefkat";
218 $fetchcontent = fopen ($abfrageurl,"r");
219 while (!feof($fetchcontent)) {
220     $savecontent = fread($fetchcontent , 4096);
221 }
222 fclose($fetchcontent);
223 $xml_prices = new SimpleXMLElement($savecontent);
224 foreach($xml_prices->Kontingent as $preise) {
225     if ((int)$preise['BelegtAnzahl'] == 0) {
226         $preis = $preise->preis_4W . '_EUR';
227
228         if ($preis != 0) {
229             ?>
230             <tr class="vak_preistr2">
231                 <td width="300px">
232                     <?php
233                         if ($java == 1) {
234                             ?>
235                             <a href="vakanzen.php?oref=<?php echo $objektreferenz ; ?>&okat=<?php
236 echo $objektkategorie ; ?>&date=<?php echo $woche1_start ;
237 ?>&days=28&anbieter=<?php echo $anbieter ; ?>&p=<?php echo
238 $preis ; ?>&view=1&sid=<?php echo $sessionid ; ?>"><?php echo
239 $woche1_start . '_bis_' . $woche4_ende ; ?></a>
240 <?php
241     }
242     else {
243         ?>
244         <a href="vakanzen-nj.php?oref=<?php echo $objektreferenz ;
245 ?>&okat=<?php echo $objektkategorie ; ?>&date=<?php echo $woche1_start ;
246 ?>&days=28&anbieter=<?php echo $anbieter ; ?>&p=<?php echo
247 $preis ; ?>&view=1&sid=<?php echo $sessionid ; ?>"><?php echo
248 $woche1_start . '_bis_' . $woche4_ende ; ?></a>
249 <?php
250     }
251     ?>
252 </td>
253 <td width="100px">
254     <?php
255         echo $preise->preis_4W . '&nbsp;EUR' ;
256     ?>
257 </td>
258 <td width="100px">
259     28 Tage
260 </td>
261 </tr>
262 <?php
263     }
264 }
265 }
266 ?>

```

267 </table>

G.9. Preisausgabe Private Objekte

Für Angebote von privat werden derzeit noch keine Preise erfasst, da eine Buchung mit dem Vermieter persönlich ausgemacht werden muss. In Zukunft könnten jedoch auch im privaten Bereich Preise angeboten werden.

G.10. Genereller Variablencheck

Listing G.8: Quelltext Überprüfungsroutine für Eingaben im Frontend

```

1 <?php
2 /**
3  * Variablen Check.php
4  * sammelt von der Index.php kommende Variablen ein und
5  * Überprüft übergebene POST Variablen auf Existenz
6  * Und stellt sie als direkte Variablen zur Verfügung
7  * Schaltet zusätzlich die entspr. Suchvariante
8  */
9 $java = 1;
10 if ($_POST['noscript'] == 1) {
11     $java = 0;
12 }
13 $_SESSION['error'] = '';
14 $_SESSION['suchwunsch'] = '';
15
16 $zeichen_illegal = array("<", ">");
17 $sprung = 0;
18 $sessionid = $_REQUEST['sid'];
19 /* Suchstatus setzen */
20 if (isset($_POST['suche'])) {
21     $suche = 1;
22 }
23 else {
24     $suche = 0;
25 }
26
27 if ($suche == 1) {
28     /* Alle Datensätze zur Auswahl aus der Datenbank lesen */
29     $querymaxentries = true;
30     /* Direkte Objektsuche */
31     /* Objekt Direktsuche checken und kürzen.
32     Nur angewendet im Dokument wo inkludiert */
33     if (isset($_POST['objektcode'])) {
34         if (strlen($_POST['objektcode']) > 5 AND strlen($_POST['objektcode']) < 40) {
35             $objektcode = str_replace($zeichen_illegal, "_", $_POST['objektcode']);
36             $objektcode = substr($objektcode, 0, 40);
37             $sprung = 1;
38         }
39     }
40     /* Ende Direkte Objektsuche */
41     /* Übernahme allgemeiner Parameter in Session */
42     /* *L1 = Landescode */
43     $_SESSION['allgemein'][0][0] = NULL;
44     if (isset($_POST['L1']) AND $_POST['L1'] != '0' AND strlen($_POST['L1']) < 4) {
45         $_SESSION['allgemein'][0][0] = $_POST['L1'];
46     }
47     else {
48         $_SESSION['allgemein'][0][0] = NULL;

```

```

49  }
50
51  /* *L2 = Regionscode* */
52  $_SESSION['allgemein'][0][1] = NULL;
53  if (isset($_POST['L2']) AND is_numeric($_POST['L2']) AND
54  strlen($_POST['L2']) == 4) {
55      $_SESSION['allgemein'][0][1] = $_POST['L2'];
56  }
57  else {
58      $_SESSION['allgemein'][0][1] = NULL;
59  }
60
61  /* *L3 = Ortscode* */
62  $_SESSION['allgemein'][0][2] = NULL;
63  if (isset($_POST['L3']) AND is_numeric($_POST['L3']) AND
64  strlen($_POST['L3']) == 7) {
65      $_SESSION['allgemein'][0][2] = $_POST['L3'];
66  }
67  else {
68      $_SESSION['allgemein'][0][2] = NULL;
69  }
70
71  /* *L4 = Anreisedatum* */
72  $_SESSION['allgemein'][1][0] = NULL;
73  if (isset($_POST['L4'])) {
74      if (strlen($_POST['L4']) < 1 OR strlen($_POST['L4']) > 10) {
75          $_SESSION['allgemein'][1][0] = date("d").'.'.date("m").'.'.date("Y");
76          $_SESSION['objekt'][2] = date("d").'.'.date("m").'.'.date("Y");
77      }
78      else {
79          $_SESSION['allgemein'][1][0] = $_POST['L4'];
80          $_SESSION['objekt'][2] = $_POST['L4'];
81      }
82  }
83
84  /* *L5 = Aufenthaltsdauer* */
85  $_SESSION['allgemein'][1][1] = NULL;
86  if (isset($_POST['L5']) AND is_numeric($_POST['L5']) AND
87  strlen($_POST['L5']) <= 2) {
88      $_SESSION['allgemein'][1][1] = $_POST['L5'];
89      $_SESSION['objekt'][3] = $_POST['L5'];
90  }
91
92  /* *L6 = Personenanzahl* */
93  /* Checken der Stringlänge, max 2 Chars erlaubt */
94  $_SESSION['allgemein'][2][0] = NULL;
95  if (isset($_POST['L6'])) {
96      if (strlen($_POST['L6']) > 2 AND $_POST['L6'] != '' AND $_POST['L6'] != 'kA') {
97          $_SESSION['allgemein'][2][0] = 2;
98      }
99      else {
100          $_SESSION['allgemein'][2][0] = $_POST['L6'];
101      }
102  }
103
104  /* *L7 = Zimmeranzahl* */
105  $_SESSION['allgemein'][2][1] = NULL;
106  if (isset($_POST['L7'])) {
107      if (strlen($_POST['L7']) > 2 AND $_POST['L7'] != '' AND $_POST['L7'] != 'kA') {
108          $_SESSION['allgemein'][2][1] = 2;
109      }
110      else {
111          $_SESSION['allgemein'][2][1] = $_POST['L7'];
112      }
113  }
114

```

```

115  /* *L8 = Schlafraumanzahl* */
116  $_SESSION['allgemein'][2][2] = NULL;
117  if (isset($_POST['L8'])) {
118      if (strlen($_POST['L8']) > 2 AND $_POST['L8'] != '' AND $_POST['L8'] != 'kA') {
119          $_SESSION['allgemein'][2][2] = 2;
120      }
121      else {
122          $_SESSION['allgemein'][2][2] = $_POST['L8'];
123      }
124  }
125  /* Übernahme Merkmale in Session */
126  /* *M1 = Haustier* */
127  $_SESSION['allgemein'][3][0] = NULL;
128  if ($_POST['M1'] == 1) {
129      $_SESSION['allgemein'][3][0] = 1;
130  }
131
132  /* *M2 = Rollstuhlgeeignet* */
133  $_SESSION['allgemein'][3][1] = NULL;
134  if (isset($_POST['M2']) AND $_POST['M2'] == '1') {
135      $_SESSION['allgemein'][3][1] = 1;
136  }
137
138  /* *M3 = Waschmaschine* */
139  $_SESSION['allgemein'][3][2] = NULL;
140  if (isset($_POST['M3']) AND $_POST['M3'] == '1') {
141      $_SESSION['allgemein'][3][2] = 1;
142  }
143
144  /* *M4 = Geschirrspüler* */
145  $_SESSION['allgemein'][3][3] = NULL;
146  if (isset($_POST['M4']) AND $_POST['M4'] == '1') {
147      $_SESSION['allgemein'][3][3] = 1;
148  }
149
150  /* *M5 = TV* */
151  $_SESSION['allgemein'][3][4] = NULL;
152  if (isset($_POST['M5']) AND $_POST['M5'] == '1') {
153      $_SESSION['allgemein'][3][4] = 1;
154  }
155
156  /* *M6 = Pool* */
157  $_SESSION['allgemein'][3][5] = NULL;
158  if (isset($_POST['M6']) AND $_POST['M6'] == '1') {
159      $_SESSION['allgemein'][3][5] = 1;
160  }
161
162  /* *M7 = Golfen* */
163  $_SESSION['allgemein'][3][6] = NULL;
164  if (isset($_POST['M7']) AND $_POST['M7'] == '1') {
165      $_SESSION['allgemein'][3][6] = 1;
166  }
167
168  /* *M8 = Kamin* */
169  $_SESSION['allgemein'][3][7] = NULL;
170  if (isset($_POST['M8']) AND $_POST['M8'] == '1') {
171      $_SESSION['allgemein'][3][7] = 1;
172  }
173
174  /* *M9 = Entfernung Angeln* */
175  $_SESSION['allgemein'][3][8] = NULL;
176  if (isset($_POST['M9']) AND is_numeric($_POST['M9']) AND
177  strlen($_POST['M9']) >=3) {
178      if (strlen($_POST['M9']) < 6) {
179          $_SESSION['allgemein'][3][8] = $_POST['M9'];
180      }

```

```

181 }
182
183 /* *M10 = Entfernung Bademöglichkeit* */
184 $_SESSION['allgemein'][3][9] = NULL;
185 if (isset($_POST['M10']) AND is_numeric($_POST['M10']) AND
186 strlen($_POST['M10']) >=3) {
187     if (strlen($_POST['M10']) < 6) {
188         $_SESSION['allgemein'][3][9] = $_POST['M10'];
189     }
190 }
191
192 /* *M11 = Entfernung Meer* */
193 $_SESSION['allgemein'][3][10] = NULL;
194 if (isset($_POST['M11']) AND is_numeric($_POST['M11']) AND
195 strlen($_POST['M11']) >=3) {
196     if (strlen($_POST['M11']) < 6) {
197         $_SESSION['allgemein'][3][10] = $_POST['M11'];
198     }
199 }
200
201 /* *M12 = Entfernung Skilifte* */
202 $_SESSION['allgemein'][3][11] = NULL;
203 if (isset($_POST['M12']) AND is_numeric($_POST['M12']) AND
204 strlen($_POST['M12']) >=3) {
205     if (strlen($_POST['M12']) < 6) {
206         $_SESSION['allgemein'][3][11] = $_POST['M12'];
207     }
208 }
209 /* Übernahme Extraoptionen in Session */
210 /* *E1 = Sauna* */
211 $_SESSION['allgemein'][4][0] = NULL;
212 if (isset($_POST['E1']) AND $_POST['E1'] == '1') {
213     $_SESSION['allgemein'][4][0] = 1;
214 }
215
216 /* *E2 = Solarium* */
217 $_SESSION['allgemein'][4][1] = NULL;
218 if (isset($_POST['E2']) AND $_POST['E2'] == '1') {
219     $_SESSION['allgemein'][4][1] = 1;
220 }
221
222 /* *E3 = Wellness* */
223 $_SESSION['allgemein'][4][2] = NULL;
224 if (isset($_POST['E3']) AND $_POST['E3'] == '1') {
225     $_SESSION['allgemein'][4][2] = 1;
226 }
227
228 /* *E4 = Whirlpool* */
229 $_SESSION['allgemein'][4][3] = NULL;
230 if (isset($_POST['E4']) AND $_POST['E4'] == '1') {
231     $_SESSION['allgemein'][4][3] = 1;
232 }
233
234 /* *E5 = Schwimmbad* */
235 $_SESSION['allgemein'][4][4] = NULL;
236 if (isset($_POST['E5']) AND $_POST['E5'] == '1') {
237     $_SESSION['allgemein'][4][4] = 1;
238 }
239
240 /* *E6 = Badelandschaft* */
241 $_SESSION['allgemein'][4][5] = NULL;
242 if (isset($_POST['E6']) AND $_POST['E6'] == '1') {
243     $_SESSION['allgemein'][4][5] = 1;
244 }
245
246 /* *E7 = Boot* */

```



```

247 $_SESSION['allgemein'][4][6] = NULL;
248 if (isset($_POST['E7']) AND $_POST['E7'] == '1') {
249     $_SESSION['allgemein'][4][6] = 1;
250 }
251
252 /* *E8 = Garage* */
253 $_SESSION['allgemein'][4][7] = NULL;
254 if (isset($_POST['E8']) AND $_POST['E8'] == '1') {
255     $_SESSION['allgemein'][4][7] = 1;
256 }
257
258 /* *E9 = Kinderbetreuung* */
259 $_SESSION['allgemein'][4][8] = NULL;
260 if (isset($_POST['E9']) AND $_POST['E9'] == '1') {
261     $_SESSION['allgemein'][4][8] = 1;
262 }
263
264 $_SESSION['allgemein'][5][0] = NULL;
265 if (isset($_POST['fhprivat']) AND $_POST['fhprivat'] == 'privat') {
266     $_SESSION['allgemein'][5][0] = $_POST['fhprivat'];
267 }
268
269 /* wunschziel nur für Ajax im Suchformular nötig, deshalb hier auf 0 */
270 if (isset($_POST['wunschziel'])) {
271     $wunschziel = 0;
272 }
273
274 /* Direktsuche checken und kürzen, keine Session,
275 da nur im Dokument wo includiert benutzt */
276 if (isset($_POST['direktsuche'])) {
277     $direktsuche = $_POST['direktsuche'];
278     if (strlen($direktsuche) > 3 AND $direktsuche != 'Wunschziel') {
279         /* *im POST ist die Codierung nach Land, Region oder Ort
280         aus der Tabelle enthalten* */
281         $direktsuche = str_replace($zeichen_illegal, "_", $_POST['direktsuche']);
282         $direktsuche = substr($direktsuche, 0, 100);
283         /* *Auswahlwerte kommen direkt als Zifferncodierung
284         aus der Tabelle und brauchen nur noch eingesetzt werden* */
285         if (strlen($direktsuche) <= 3) {
286             $_SESSION['allgemein'][0][0] = $direktsuche;
287             $_SESSION['allgemein'][0][1] = 'kA';
288             $_SESSION['allgemein'][0][2] = 'kA';
289         }
290         if (strlen($direktsuche) == 4) {
291             $query = "SELECT_LandID_FROM_ref_region_WHERE
292             IDRegion=_$direktsuche_LIMIT_1";
293             $result = mysql_query($query);
294             while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
295                 $direktland = $line['LandID'];
296             }
297             $_SESSION['allgemein'][0][0] = $direktland;
298             $_SESSION['allgemein'][0][1] = $direktsuche;
299             $_SESSION['allgemein'][0][2] = 'kA';
300         }
301         if (strlen($direktsuche) == 7) {
302             $query = "SELECT_RegionID,_LandID_FROM_ref_ort_WHERE
303             IDOrt=_$direktsuche_LIMIT_1";
304             $result = mysql_query($query);
305             while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
306                 $direktland = $line['LandID'];
307                 $direktregion = $line['RegionID'];
308             }
309             $_SESSION['allgemein'][0][0] = $direktland;
310             $_SESSION['allgemein'][0][1] = $direktregion;
311             $_SESSION['allgemein'][0][2] = $direktsuche;
312         }

```

```

313     }
314 }
315
316 if ($sprung == 1) {
317     if (strpos($objektcode, '.')) {
318         $objektcode_array = explode(".", $objektcode);
319         $teile = count($objektcode_array);
320         $objektreferenz = '';
321         $objektkategorie = '';
322         for ($i = 0; $i < $teile - 1; $i++) {
323             $objektreferenz_array[$i] = $objektcode_array[$i];
324         }
325         $objektreferenz = implode(".", $objektreferenz_array);
326         $objektkategorie = $objektcode_array[$teile - 1];
327     }
328
329     $query = "SELECT_ObjReferenz, _Kategorie, _IDAnbieter FROM
330     _$_$tabelle.obj_stammdaten WHERE_ObjReferenz LIKE
331     _$_$objektreferenz' AND_Kategorie LIKE_' $objektkategorie' _LIMIT_1";
332     $result = mysql_query($query);
333     if (mysql_num_rows($result) < 1) {
334         $_SESSION['error'] = "Kein_zum_Code_passendes_Objekt_gefunden";
335         Header("Location:index.php");
336         exit();
337     }
338     else {
339         while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
340             $objektreferenz = $line['ObjReferenz'];
341             $objektkategorie = $line['Kategorie'];
342             $anbieter = $line['IDAnbieter'];
343         }
344         $_SESSION['objekt'][0] = $objektreferenz;
345         $_SESSION['objekt'][1] = $objektkategorie;
346         $_SESSION['objekt'][4] = $anbieter;
347         if ($java == 1) {
348             Header("Location:details_ferienhaus.php?direkt=1&sid=$sessionid");
349         }
350         else {
351             Header("Location:details_ferienhaus-nj.php?direkt=1&sid=$sessionid");
352         }
353         exit();
354     }
355 }
356 }
357
358 /* Weitere Abfragen bei Seitennavigation, übergeben als POST oder GET,
359 deshalb REQUEST */
360 if (isset($_REQUEST['eintrakt'])) {
361     if (strlen($_REQUEST['eintrakt']) > 3) {
362         $eintrakt = str_replace($zeichen_illegal, "_", $_REQUEST['eintrakt']);
363         $eintrakt = substr($eintrakt, 0, 3);
364         $_SESSION['navigation'][0] = $eintrakt;
365     }
366     else {
367         $_SESSION['navigation'][0] = $_REQUEST['eintrakt'];
368     }
369 }
370 else {
371     $_SESSION['navigation'][0] = 0;
372 }
373
374 if (isset($_REQUEST['seitenstart'])) {
375     if (strlen($_REQUEST['seitenstart']) > 3) {
376         $seitenstart = str_replace($zeichen_illegal, "_", $_REQUEST['seitenstart']);
377         $seitenstart = substr($eintrakt, 0, 3);
378         $_SESSION['navigation'][1] = $seitenstart;

```

```

379     }
380     else {
381         $_SESSION['navigation'][1] = $_REQUEST['seitenstart'];
382     }
383 }
384 else {
385     $_SESSION['navigation'][1] = 1;
386 }
387 ?>

```

G.11. Buchungsvorgang Teil 1

Listing G.9: Quelltext Buchen 1

```

1 <?php
2
3 /**
4  * Buchungsbereich 1
5  * Erfassung Personendaten
6  * Erfassung Zusatzleistungen
7  * Ausgabe und Verarbeitung von
8  * Buchungshinweisen
9 */
10
11 $sessionid = $_REQUEST['sid'];
12 if (!isset($_COOKIE['PHPSESSID'])) {
13     session_id($sessionid);
14 }
15 session_start();
16
17 /* Abfrageschnittstellen einbinden */
18 $pfad = getenv("DOCUMENT_ROOT");
19 $datenbankzugriff = $pfad.'/include/opendb-ferienhaus.php';
20 include $datenbankzugriff;
21 $inc_objektklasse = $pfad.'/include/class-objekte-db.php';
22 include $inc_objektklasse;
23 $inc_datumsklasse = $pfad.'/include/class-datumfkt.php';
24 include $inc_datumsklasse;
25 $inc_geldklasse = $pfad.'/include/class-geld.php';
26 include $inc_geldklasse;
27
28 /* Abfrageschnittstellen initialisieren */
29 $objektdaten = new datensaetze;
30 $moneur = new changeto-eur;
31
32 /* Objektdaten referenzieren */
33 $objektreferenz = $_SESSION['objekt'][0];
34 $objektkategorie = $_SESSION['objekt'][1];
35 $anreisedatum = $_SESSION['objekt'][2];
36 $aufenthaltsdauer = $_SESSION['objekt'][3];
37 $anbieter = $_SESSION['objekt'][4];
38
39 /* Variablen deklarieren */
40 $tabelle='usr_web31_1';
41 $fehler = $_SESSION['meldung'];
42
43 /* Personenanzahl für das Objekt ermitteln */
44 $objektdaten->personenanzahl_holen($objektreferenz, $objektkategorie);
45 $anz_personen = $objektdaten->personenanzahl;
46 $anz_kinder = $objektdaten->kinderanzahl;
47
48 /* *Tiefe zum Stammverzeichnis ermitteln* */
49 $downward = "";
50 $pfad = getenv("REQUEST_URI");

```

```

51 $Pfadteile = explode("",$Pfad);
52 for($i=count($Pfadteile)-1;$i>=1;$i--)
53 {
54     $downward = $downward . " ../";
55 }
56
57 /* Spezielle Funktionen deklarieren */
58 /* ***Funktion Interhome Zusatzleistungen*** */
59 function zusatzleistungen_interhome ($pfad, $anbieter, $objektreferenz,
60 $objektkategorie, $anreisedatum, $aufenthaltsdauer, $fehler) {
61
62     /* Abfrageschnittstellen initialisieren */
63     $objektdaten = new datensaetze;
64     $datumsfkt = new zeitfunktionen;
65     /* ----- Interhome SOAP Header laden ----- */
66     $interhome_soap_connect = $pfad.'/include/interhome_soapconnect.php';
67     include $interhome_soap_connect;
68
69     /* Ermittlung Anreisedatum im Unix Format und YYYY-MM-DD */
70     $ih_objektcode = $objektreferenz.'.'. $objektkategorie;
71     $datumsfkt->datum_deisoun($anreisedatum);
72     $anreisedatum_iso = $datumsfkt->datum_iso;
73     $anreisedatum_unix = $datumsfkt->datum_unix;
74
75     /* Ermittlung Abreisedatum im Unix Format und YYYY-MM-DD */
76     $duration = $aufenthaltsdauer*86400;
77     $abreisedatum_unix = $anreisedatum_unix+$duration;
78     $abreisedatum_iso = strftime("%Y-%m-%d", $abreisedatum_unix);
79
80     /* Interhome SOAP Abfrage (PriceRqt) */
81     $params = array('So' => '5059', 'Nref' => $ih_objektcode, 'CheckIn' =>
82     $anreisedatum_iso, 'CheckOut' => $abreisedatum_iso,
83     'ExtraBed' => '', 'InsAins' => 'Y', 'Environment' => '0');
84     $result = $client->__soapCall("PriceRqt",array('parameters'
85 =>$params),null,$header);
86
87     $anzahlungbis_iso = (string)$result->PriceRqtResult->ExpirationDatePrePayment;
88     $datumsfkt->datum_isode($anzahlungbis_iso);
89     $anzahlungbis = $datumsfkt->datum_de;
90     $restbetragbis_iso = (string)$result->PriceRqtResult->ExpirationDateResidue;
91     $datumsfkt->datum_isode($restbetragbis_iso);
92     $restbetragbis = $datumsfkt->datum_de;
93
94     /* Verteilung der Werte aus $result */
95     $preise_objekt[0] = array((string)$result->PriceRqtResult->PriceRent,
96 (string)$result->PriceRqtResult->Currency);
97     $preise_objekt[1] = array((string)$result->PriceRqtResult->PricePrePayment,
98 (string)$result->PriceRqtResult->Currency);
99     $zahlungstermine = array($anzahlungbis, $restbetragbis);
100    $rrkv[0] = array(0 => (string)$result->PriceRqtResult->InsExtraDesc1, 1 =>
101 (string)$result->PriceRqtResult->PriceInsExtra1, 2 =>
102 (string)$result->PriceRqtResult->Currency);
103    $rrkv[1] = array(0 => (string)$result->PriceRqtResult->InsExtraDesc2, 1 =>
104 (string)$result->PriceRqtResult->PriceInsExtra2, 2 =>
105 (string)$result->PriceRqtResult->Currency);
106
107    /* Zusatzleistungen ermitteln */
108    $objektdaten->zusatzleistungen_holen_getrennt($anbieter, $objektreferenz,
109 $objektkategorie);
110    $zubs_incl = $objektdaten->zubs_incl;
111    $zubs_vo = $objektdaten->zubs_vo;
112    $zubs_zuvo = $objektdaten->zubs_zuvo;
113    $zubs_incl_eintraege = count($zubs_incl);
114    $zubs_vo_eintraege = count($zubs_vo);
115    $zubs_zuvo_eintraege = count($zubs_zuvo);
116

```

```

117  /* Ausgabe */
118  ?>
119  <strong>Reise R&#252;cktritts Kostenschutz:</strong><br>
120  <table>
121    <tr>
122      <td>
123        <?php
124          if (isset($_SESSION['zusatz'][0][0]) AND ($_SESSION['zusatz'][0][0] ==
125            'INSA' OR $_SESSION['zusatz'][0][0] == 'INSB')) {
126            ?>
127            <input type="radio" name="rrkv" value="KEIN_Kein_Kostenschutz_0_EUR">
128            <?php
129              }
130            else {
131              ?>
132              <input type="radio" name="rrkv" value="KEIN_Kein_Kostenschutz_0_EUR"
133                checked="checked">
134              <?php
135                }
136              ?>
137            </td>
138            <td>
139              Kein Reise R&#252;cktritts Kostenschutz
140            </td>
141          </tr>
142          <tr>
143            <td>
144              <?php
145                if (isset($_SESSION['zusatz'][0][0]) AND $_SESSION['zusatz'][0][0] ==
146                  'INSB') {
147                  ?>
148                  <input type="radio" name="rrkv" value="INSB-<?php_echo_&#36;rrkv[0][0];
149                    &#36;rrkv[0][1]; &#36;rrkv[0][2]; &#36;rrkv[0][3]; &#36;rrkv[0][4]; &#36;rrkv[0][5]; &#36;rrkv[0][6]; &#36;rrkv[0][7]; &#36;rrkv[0][8]; &#36;rrkv[0][9]; &#36;rrkv[0][10]; &#36;rrkv[0][11]; &#36;rrkv[0][12]; &#36;rrkv[0][13]; &#36;rrkv[0][14]; &#36;rrkv[0][15]; &#36;rrkv[0][16]; &#36;rrkv[0][17]; &#36;rrkv[0][18]; &#36;rrkv[0][19]; &#36;rrkv[0][20]; &#36;rrkv[0][21]; &#36;rrkv[0][22]; &#36;rrkv[0][23]; &#36;rrkv[0][24]; &#36;rrkv[0][25]; &#36;rrkv[0][26]; &#36;rrkv[0][27]; &#36;rrkv[0][28]; &#36;rrkv[0][29]; &#36;rrkv[0][30]; &#36;rrkv[0][31]; &#36;rrkv[0][32]; &#36;rrkv[0][33]; &#36;rrkv[0][34]; &#36;rrkv[0][35]; &#36;rrkv[0][36]; &#36;rrkv[0][37]; &#36;rrkv[0][38]; &#36;rrkv[0][39]; &#36;rrkv[0][40]; &#36;rrkv[0][41]; &#36;rrkv[0][42]; &#36;rrkv[0][43]; &#36;rrkv[0][44]; &#36;rrkv[0][45]; &#36;rrkv[0][46]; &#36;rrkv[0][47]; &#36;rrkv[0][48]; &#36;rrkv[0][49]; &#36;rrkv[0][50]; &#36;rrkv[0][51]; &#36;rrkv[0][52]; &#36;rrkv[0][53]; &#36;rrkv[0][54]; &#36;rrkv[0][55]; &#36;rrkv[0][56]; &#36;rrkv[0][57]; &#36;rrkv[0][58]; &#36;rrkv[0][59]; &#36;rrkv[0][60]; &#36;rrkv[0][61]; &#36;rrkv[0][62]; &#36;rrkv[0][63]; &#36;rrkv[0][64]; &#36;rrkv[0][65]; &#36;rrkv[0][66]; &#36;rrkv[0][67]; &#36;rrkv[0][68]; &#36;rrkv[0][69]; &#36;rrkv[0][70]; &#36;rrkv[0][71]; &#36;rrkv[0][72]; &#36;rrkv[0][73]; &#36;rrkv[0][74]; &#36;rrkv[0][75]; &#36;rrkv[0][76]; &#36;rrkv[0][77]; &#36;rrkv[0][78]; &#36;rrkv[0][79]; &#36;rrkv[0][80]; &#36;rrkv[0][81]; &#36;rrkv[0][82]; &#36;rrkv[0][83]; &#36;rrkv[0][84]; &#36;rrkv[0][85]; &#36;rrkv[0][86]; &#36;rrkv[0][87]; &#36;rrkv[0][88]; &#36;rrkv[0][89]; &#36;rrkv[0][90]; &#36;rrkv[0][91]; &#36;rrkv[0][92]; &#36;rrkv[0][93]; &#36;rrkv[0][94]; &#36;rrkv[0][95]; &#36;rrkv[0][96]; &#36;rrkv[0][97]; &#36;rrkv[0][98]; &#36;rrkv[0][99]; &#36;rrkv[0][100]; &#36;rrkv[0][101]; &#36;rrkv[0][102]; &#36;rrkv[0][103]; &#36;rrkv[0][104]; &#36;rrkv[0][105]; &#36;rrkv[0][106]; &#36;rrkv[0][107]; &#36;rrkv[0][108]; &#36;rrkv[0][109]; &#36;rrkv[0][110]; &#36;rrkv[0][111]; &#36;rrkv[0][112]; &#36;rrkv[0][113]; &#36;rrkv[0][114]; &#36;rrkv[0][115]; &#36;rrkv[0][116]; &#36;rrkv[0][117]; &#36;rrkv[0][118]; &#36;rrkv[0][119]; &#36;rrkv[0][120]; &#36;rrkv[0][121]; &#36;rrkv[0][122]; &#36;rrkv[0][123]; &#36;rrkv[0][124]; &#36;rrkv[0][125]; &#36;rrkv[0][126]; &#36;rrkv[0][127]; &#36;rrkv[0][128]; &#36;rrkv[0][129]; &#36;rrkv[0][130]; &#36;rrkv[0][131]; &#36;rrkv[0][132]; &#36;rrkv[0][133]; &#36;rrkv[0][134]; &#36;rrkv[0][135]; &#36;rrkv[0][136]; &#36;rrkv[0][137]; &#36;rrkv[0][138]; &#36;rrkv[0][139]; &#36;rrkv[0][140]; &#36;rrkv[0][141]; &#36;rrkv[0][142]; &#36;rrkv[0][143]; &#36;rrkv[0][144]; &#36;rrkv[0][145]; &#36;rrkv[0][146]; &#36;rrkv[0][147]; &#36;rrkv[0][148]; &#36;rrkv[0][149]; &#36;rrkv[0][150]; &#36;rrkv[0][151]; &#36;rrkv[0][152]; &#36;rrkv[0][153]; &#36;rrkv[0][154]; &#36;rrkv[0][155]; &#36;rrkv[0][156]; &#36;rrkv[0][157]; &#36;rrkv[0][158]; &#36;rrkv[0][159]; &#36;rrkv[0][160]; &#36;rrkv[0][161]; &#36;rrkv[0][162]; &#36;rrkv[0][163]; &#36;rrkv[0][164]; &#36;rrkv[0][165]; &#36;rrkv[0][166]; &#36;rrkv[0][167]; &#36;rrkv[0][168]; &#36;rrkv[0][169]; &#36;rrkv[0][170]; &#36;rrkv[0][171]; &#36;rrkv[0][172]; &#36;rrkv[0][173]; &#36;rrkv[0][174]; &#36;rrkv[0][175]; &#36;rrkv[0][176]; &#36;rrkv[0][177]; &#36;rrkv[0][178]; &#36;rrkv[0][179]; &#36;rrkv[0][180]; &#36;rrkv[0][181]; &#36;rrkv[0][182]; &#36;rrkv[0][183]; &#36;rrkv[0][184]; &#36;rrkv[0][185]; &#36;rrkv[0][186]; &#36;rrkv[0][187]; &#36;rrkv[0][188]; &#36;rrkv[0][189]; &#36;rrkv[0][190]; &#36;rrkv[0][191]; &#36;rrkv[0][192]; &#36;rrkv[0][193]; &#36;rrkv[0][194]; &#36;rrkv[0][195]; &#36;rrkv[0][196]; &#36;rrkv[0][197]; &#36;rrkv[0][198]; &#36;rrkv[0][199]; &#36;rrkv[0][200]; &#36;rrkv[0][201]; &#36;rrkv[0][202]; &#36;rrkv[0][203]; &#36;rrkv[0][204]; &#36;rrkv[0][205]; &#36;rrkv[0][206]; &#36;rrkv[0][207]; &#36;rrkv[0][208]; &#36;rrkv[0][209]; &#36;rrkv[0][210]; &#36;rrkv[0][211]; &#36;rrkv[0][212]; &#36;rrkv[0][213]; &#36;rrkv[0][214]; &#36;rrkv[0][215]; &#36;rrkv[0][216]; &#36;rrkv[0][217]; &#36;rrkv[0][218]; &#36;rrkv[0][219]; &#36;rrkv[0][220]; &#36;rrkv[0][221]; &#36;rrkv[0][222]; &#36;rrkv[0][223]; &#36;rrkv[0][224]; &#36;rrkv[0][225]; &#36;rrkv[0][226]; &#36;rrkv[0][227]; &#36;rrkv[0][228]; &#36;rrkv[0][229]; &#36;rrkv[0][230]; &#36;rrkv[0][231]; &#36;rrkv[0][232]; &
```

```

183         </td>
184         <td>
185             <?php echo $rrkv[1][0]. '&nbsp;'; $rrkv[1][1]. '&nbsp;'; $rrkv[1][2]; ?>
186         </td>
187     </tr>
188 </table>
189 <br>
190 <strong>Im Preis enthalten</strong><br>
191 <?php
192 if ($zubs_incl[0][0] != '-') {
193     for ($i = 0; $i < $zubs_incl_eintraege; $i++) {
194         echo '<li>'. utf8_encode($zubs_incl[$i][0]). '</li>';
195     }
196 }
197 else { echo '<li>Es liegen keine im Preis enthaltenen Leistungen vor</li>'; }
198 ?>
199 <br>
200 <strong>Vor Ort zu bezahlen:</strong><br>
201 <?php
202 $preisvo = 0;
203
204 if ($zubs_vo[0][0] != '-') {
205     for ($i = 0; $i < $zubs_vo_eintraege; $i++) {
206         $waehrung = $zubs_vo[$i][2];
207         echo '<li>'. utf8_encode($zubs_vo[$i][0]). '&nbsp;'.
208             $zubs_vo[$i][1]. '&nbsp;'. $zubs_vo[$i][2]. '&nbsp;'. $zubs_vo[$i][4]. '</li>';
209         $_SESSION['zusatz'][$i+31] = array(0, $zubs_vo[$i][0],
210             $zubs_vo[$i][1], $zubs_vo[$i][2], 1, $zubs_vo[$i][4]);
211     }
212 }
213 else { echo '<li>Es liegen keine vor Ort zu bezahlenden Leistungen vor</li>'; }
214 ?>
215 <br>
216 <strong>Zus&#228;tzlich buchbar und vor Ort zu bezahlen:</strong><br>
217 <?php
218 /* Struktur $zubs_zuvo -> Array=> (Leistung=>Array(0=> Bezeichnung, 1=>Preis,
219 2=>Waehrung, 3=>Serviceart, 4=>Berechnung)) */
220 /* Leistungen zum Objekt vorhanden?*/
221 if ($zubs_zuvo[0][0] != '-') {
222     for ($i = 0; $i < $zubs_zuvo_eintraege; $i++) {
223         /* Portierung array, verständlichere Form für die Anwendung*/
224         $zu_leistungsart = $zubs_zuvo[$i][0];
225         $zu_kosten = $zubs_zuvo[$i][1];
226         $zu_waehrung = $zubs_zuvo[$i][2];
227         $zu_serviceart = $zubs_zuvo[$i][3];
228         $zu_berechnung1 = $zubs_zuvo[$i][4];
229         $zu_berechnung2 = str_replace("-", "", $zubs_zuvo[$i][4]);
230         $sess_zusleist = $_SESSION['zusatz'];
231         /* Initialisieren Variable für bestimmung ob Leistung vorher schon ausgewählt*/
232         $mmchecked = 0;
233         /* Ausführung wenn Zusatzleistung Geld kostet */
234         if ($zu_kosten != 0.00 AND $zu_kosten != '') {
235
236             $query = "SELECT IDZusatz FROM ref_zusatzleistungen WHERE Beschreibung LIKE
237             ' $zu_leistungsart ' LIMIT 1";
238             $result = mysql_query($query);
239             while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
240                 $zu_lacode = $line['IDZusatz'];
241             }
242             /* Check ob Leistung bereits in Session == vorher schonmal ausgewählt */
243             foreach ($sess_zusleist AS $szusatz) {
244                 if ($szusatz[0] == $zu_lacode) {
245                     $mmchecked = 1;
246                 }
247             }
248             /* Ausgabe Checkbox für entspr Leistung */

```

```

249     $cbox_value = $zu_lacode.'-'.utf8_encode($zu_leistungsart).'-'.
250     $zu_kosten.'-'. $zu_waehrung.'-'. $zu_berechnung2;
251     $cbox_info = utf8_encode($zu_leistungsart).'&nbsp;'. $zu_kosten.'&nbsp;'.
252     $zu_waehrung.'&nbsp;'. $zu_berechnung1;
253     if ($mmchecked == 1) {
254         echo '<li><input type="checkbox" name="'. $i.' "_value="'. $cbox_value.'"
255         checked="checked"></input>'. $cbox_info.'</li>';
256     }
257     else {
258         echo '<li><input type="checkbox" name="'. $i.' "_value="'. $cbox_value.
259         '"></input>'. $cbox_info.'</li>';
260     }
261 }
262 else {
263     echo '<li>'.utf8_encode($zu_leistungsart).'&nbsp;kostenlos<li>';
264 }
265 }
266 }
267 else { echo '<li>Es liegen keine zubuchbaren Leistungen vor</li>'; }
268 ?>
269 <div class="headline">Preise und Zahlungsdetails</div>
270 <?php
271 if ($preise_objekt[0][0] <= $preise_objekt[1][0]) {
272     echo 'Objektpreis:'. $preise_objekt[0][0]. '&nbsp;'. $preise_objekt[0][1].
273     '&nbsp;f&#252;r&nbsp;'. $aufenthaltsdauer.'&nbsp;Tage<br>';
274 }
275 else {
276     echo 'Objektpreis:'. $preise_objekt[0][0]. '&nbsp;'. $preise_objekt[0][1].
277     '&nbsp;f&#252;r&nbsp;'. $aufenthaltsdauer.'&nbsp;Tage<br>';
278     echo 'Anzahlung:'. $preise_objekt[1][0]. '&nbsp;'. $preise_objekt[1][1]. '<br>';
279 }
280
281 if ($anzahlungbis == '01.01.0001') {
282     echo 'Der Betrag muss bis zum'. $restbetragbis.' beglichen werden.<br>';
283 }
284 else {
285     echo 'Die Anzahlung ist bis zum'. $anzahlungbis.' f&#228;llig!<br>';
286     echo 'Der Restbetrag muss bis zum'. $restbetragbis.' beglichen werden.<br>';
287 }
288 ?>
289 <?php
290 /* Status f&#252;r Buchungsm&#246;glichkeit "auf Rechnung" festlegen */
291 $aktuelles_datum = date("U"); //Unix timestamp vom aktuellen Datum
292 $differenz_datum_29d = $aktuelles_datum+2505600; //Datum aktuell +29 Tage
293
294 $datumsfkt->datum_deisoun($anreisedatum);
295 $aufenthalt_iso = $datumsfkt->datum_iso;
296 $aufenthalt_unix = $datumsfkt->datum_unix;
297
298 if ($differenz_datum_29d > $aufenthalt_unix) {
299     $button = 'disabled="disabled"';
300 }
301 else {
302     $button = '';
303 }
304 ?>
305 <br>
306 Zahlungsart:<br>
307 <!-- $button wird geschaltet um Bezahlung per Rechnung zu aktivieren oder
308 deaktivieren -->
309 <?php
310 if ($_SESSION['payment'][0] == 2 OR $_SESSION['payment'][0] == '') {
311     ?>
312     <input type="radio" name="zahlungsart" value="2" checked="checked">
313     </input>Kreditkarte (Es sind Visa oder Mastercard m&#246;glich)<br>
314     <?php

```

```

315     }
316     else {
317     ?>
318         <input type="radio" name="zahlungsart" value="2"></input>Kreditkarte (Es sind
319         Visa oder Mastercard m<#246;glich)<br>
320     <?php
321     }
322     if ($_SESSION['payment'][0] == 3) {
323     ?>
324         <input type="radio" name="zahlungsart" value="3" <?php echo $button; ?>
325         checked="checked"></input>Rechnung (Nur bis max 29 Tage vor Reiseantritt
326         m<#246;glich)<br>
327     <?php
328     }
329     else {
330     ?>
331         <input type="radio" name="zahlungsart" value="3" <?php echo
332         $button; ?>></input>Rechnung (Nur bis max 29 Tage vor Reiseantritt
333         m<#246;glich)<br>
334     <?php
335     }
336     ?>
337
338     <br>
339     <strong>Dieses Objekt wird bereitgestellt von Interhome.</strong><br>Wenn sie
340     dieses Objekt buchen schlie&#223;en sie einen Vertrag mit Interhome
341     ab.<br>Folgendes m<#252;ssen sie deshalb beachten:<br><br>
342     1. Bitte leisten Sie innerhalb von 7 Tagen eine Anzahlung von 10% der Mietsumme.
343     Der Restbetrag ist sp&#228;testens 28 Tage vor Mietbeginn zahlbar. Der sicherste und
344     schnellste Weg diese Anzahlung zu leisten , ist die Abwicklung &#252;ber Ihre
345     Kreditkarte: bitte nennen Sie uns Ihre Kreditkartennummer und das Verfalldatum. B
346     ei kurzfristigen Buchungen (alle Buchungen, die weniger als 28 Tage vor Mietbeginn
347     get&#228;tigt werden) ist der gesamte Mietpreis sofort f&#228;llig.<br><br>
348     F&#252;r den R&#252;cktritt von einer best&#228;tigten Buchung belastet Interhome die
349     folgenden Storno-Geb&#252;hren:<br>
350     <table>
351     <tr>
352         <td>Tage vor Reiseantritt</td><td>Storno Geb&#252;hr</td>
353     </tr>
354     <tr>
355         <td>bis 42</td><td>10% Des Mietbetrages</td>
356     </tr>
357     <tr>
358         <td>41 - 28</td><td>50% Des Mietbetrages</td>
359     </tr>
360     <tr>
361         <td>27 - 1</td><td>80% Des Mietbetrages</td>
362     </tr>
363     <tr>
364         <td>0 oder sp&#228;ter</td><td>100% Des Mietbetrages</td>
365     </tr>
366     </table>
367     <div id="agberror" style="color:#FF0000;">&#xA0;<?php echo $fehler[11];?></div>
368     <input type="checkbox" name="agb" value="Y"></input>Ich kenne die <a
369     target="_blank" href="http://www.interhome.de/ContractConditions/Default.aspx">
370     Vertragsbedingungen</a> und akzeptiere sie<br>
371 <?php
372 }
373 /* ***Ende Funktion Interhome Zusatzleistungen*** */
374
375 function zusatzleistungen_wolters ($pfad, $anbieter, $objektreferenz,
376 $objektkategorie, $anreisedatum, $aufenthaltsdauer, $fehler) {
377
378     /* Abfrageschnittstellen initialisieren */
379     $objektdaten = new datensaetze;
380     $datumsfkt = new zeitfunktionen;

```



```

381
382  /* Personenanzahl für das Objekt ermitteln */
383  $objektdaten->personenanzahl-holen($objektreferenz , $objektkategorie);
384  $anz_personen = $objektdaten->personenanzahl+1;
385
386  /* - Wolters Haustiere aus Merkmalstabelle für den "pet" Parameter abfragen - */
387  $petmerk = 0;
388  $petquery = "SELECT_Haustier_FROM_$tabelle.obj_merkmale_WHERE_IDAnbieter=
389  _$anbieter_AND_Objreferenz=_ '$objektreferenz' _AND_Kategorie=_ '$objektkategorie'
390  _LIMIT_1";
391  $petabfrage = mysql_query($petquery);
392  while($petline = mysql_fetch_array($petabfrage , MYSQL_ASSOC)) {
393      $petmerk = $petline['Haustier'];
394  }
395
396  /* Buchungshinweise ermitteln */
397  $nobuhinw = '_';
398  $hinweisquery = "SELECT_Buhinweis,_Datumvon,_Datumbis_FROM_$tabelle.obj_buhinweise
399  _WHERE_IDAnbieter=_ $anbieter_AND_Objreferenz=_ '$objektreferenz' _LIMIT_20";
400  $hinweisabfrage = mysql_query($hinweisquery);
401  $i = 0;
402  if (mysql_num_rows($hinweisabfrage) < 1) {
403      $hinweisparam[0] = array('KeineBuchungshinweise_vorhanden' , 0,0);
404      $nobuhinw = 'checked="checked"';
405  }
406  else {
407      while($hinweisline = mysql_fetch_array($hinweisabfrage , MYSQL_ASSOC)) {
408          if ($hinweisline['Datumvon'] == '0000-00-00' AND $hinweisline['Datumbis'] ==
409              '0000-00-00') {
410              $hinweisparam[$i] = array($hinweisline['Buhinweis'] ,0,0);
411          }
412          else {
413              $datumsfkt->isode($hinweisline['Datumvon']);
414              $datumvon->datum_de;
415              $datumsfkt->isode($hinweisline['Datumbis']);
416              $datumbis->datum_de;
417              $hinweisparam[$i] = array($hinweisline['Buhinweis'] , $datumvon, $datumbis);
418          }
419          $i++;
420      }
421  }
422
423  /* Zusatzleistungen ermitteln */
424  $objektdaten->zusatzleistungen-holen($anbieter , $objektreferenz , $objektkategorie);
425  $zubs = $objektdaten->zubs;
426  $zubs_eintraege = count($zubs);
427
428  /* Zeit bis zum Reiseantritt Ermitteln */
429  $datum_aktuell = mktime(0, 0, 0, date("m"), date("d"), date("Y"));
430  $datumsfkt->datum_deisoun($anreisedatum);
431  $datum_start = $datumsfkt->datum_unix;
432  $datum_6week = $datum_aktuell+3628800; //3628800 = 6 Wochen
433
434  if ($datum_6week < $datum_start) {
435      $disabled = '_';
436  }
437  else {
438      $disabled = 'disabled="disabled"';
439  }
440
441  /* Zusatzleistungen ausgeben */
442  $k = 0;
443  $pet_zusatzl = 0;
444  $mmchecked = 0;
445  if (is_array($zubs)) {
446      foreach ($zubs AS $leistung) {

```

```

447  /* Haustier als Zusatzleistung, max 1 Haustier */
448  if ($leistung['Leistung'] == 'Haustier') {
449      $pet_zusatzl = 1;
450      ?>
451      <select name="<?php_echo_{$k};_?>" size="1">
452          <?php
453              $k++;
454              $primary = 0;
455              $val = 0;
456              /* Session "zusatz" nach bereits gesetztem Wert durchsuchen */
457              foreach ($SESSION['zusatz'] AS $szusatz) {
458                  if ($szusatz[1] == utf8_encode($leistung['Leistung'])) {
459                      $primary = 1;
460                      $anzahl = $szusatz[4];
461                  }
462              }
463              for ($i = 0; $i <= 1; $i++) {
464                  $val = $i.'_'.utf8_encode($leistung['Leistung']).'_'.
465                      $leistung['Preis'].'_'.utf8_encode($leistung['Waehrung']).'_'.
466                      utf8_encode($leistung['Berechnung']);
467                  if ($primary == 0 AND $i == 0) {
468                      echo '<option_value="'. $i. '"_selected="selected"></option>';
469                  }
470                  elseif ($primary == 1 AND $i == 0){
471                      echo '<option_value="'. $i. '"></option>';
472                  }
473                  elseif ($primary == 1 AND $anzahl == $i) {
474                      echo '<option_value="'. $val. '"_selected="selected">'. $i. '</option>';
475                  }
476                  else {
477                      echo '<option_value="'. $val. '">'. $i. '</option>';
478                  }
479              }
480          ?>
481      </select>
482      <?php
483          echo utf8_encode($leistung['Leistung']).',&nbsp;'.
484              $leistung['Preis'].'&nbsp;'. $leistung['Waehrung'].
485              ',&nbsp;'.utf8_encode($leistung['Berechnung']). '<br>';
486      }
487      /* Endreinigung, nur 1 auswählbar*/
488      elseif ($leistung['Leistung'] == 'Endreinigung') {
489          ?>
490          <select name="<?php_echo_{$k};_?>" size="1">
491              <?php
492                  $k++;
493                  $primary = 0;
494                  $val = 0;
495                  foreach ($SESSION['zusatz'] AS $szusatz) {
496                      if ($szusatz[1] == utf8_encode($leistung['Leistung'])) {
497                          $primary = 1;
498                          $anzahl = $szusatz[4];
499                      }
500                  }
501                  for ($i = 0; $i <= 1; $i++) {
502                      $val = $i.'_'.utf8_encode($leistung['Leistung']).'_'.
503                          $leistung['Preis'].'_'. $leistung['Waehrung'].'_'.
504                          utf8_encode($leistung['Berechnung']);
505                      if ($primary == 0 AND $i == 0) {
506                          echo '<option_value="'. $i. '"_selected="selected"></option>';
507                      }
508                      elseif ($primary == 1 AND $i == 0){
509                          echo '<option_value="'. $i. '"></option>';
510                      }
511                      elseif ($primary == 1 AND $anzahl == $i) {
512                          echo '<option_value="'. $val. '"_selected="selected">'. $i. '</option>';

```

```

513     }
514     else {
515         echo '<option_value="' . $val . '">' . $i . '</option>';
516     }
517 }
518 ?>
519 </select>
520 <?php
521 echo utf8_encode($leistung['Leistung']).',&nbsp;'.
522 $leistung['Preis'].'&nbsp;'. $leistung['Waehrung'].'&nbsp;'.
523 utf8_encode($leistung['Berechnung']). '<br>';
524 }
525 else {
526     ?>
527     <select name="<?php_echo_$k;_?>" size="1">
528         <?php
529         $k++;
530         $primary = 0;
531         $val = 0;
532         foreach ($SESSION['zusatz'] AS $szusatz) {
533             if ($szusatz[1] == utf8_encode($leistung['Leistung'])) {
534                 $primary = 1;
535                 $anzahl = $szusatz[4];
536             }
537         }
538         for ($i = 0; $i < $anz_personen; $i++) {
539             $val = $i . ' . ' . utf8_encode($leistung['Leistung']). ' . ' .
540             $leistung['Preis'] . ' . ' . $leistung['Waehrung'] . ' . ' .
541             utf8_encode($leistung['Berechnung']);
542             if ($primary == 0 AND $i == 0) {
543                 echo '<option_value="' . $i . '" _selected="selected"></option>';
544             }
545             elseif ($primary == 1 AND $i == 0){
546                 echo '<option_value="' . $i . '"></option>';
547             }
548             elseif ($primary == 1 AND $anzahl == $i) {
549                 echo '<option_value="' . $val . '" _selected="selected">' . $i . '</option>';
550             }
551             else {
552                 echo '<option_value="' . $val . '">' . $i . '</option>';
553             }
554         }
555     ?>
556     </select>
557     <?php
558     echo utf8_encode($leistung['Leistung']).',&nbsp;'.
559     $leistung['Preis'].'&nbsp;'. $leistung['Waehrung'].
560     ',&nbsp;'. utf8_encode($leistung['Berechnung']). '<br>';
561 }
562 $i++;
563 }
564 }
565
566 /* Haustier aus den Merkmalen ziehen wenn nicht als Zusatzleistung vorhanden */
567 if ($petmerkmal != 0 AND $pet_zusatzl == 0) {
568     ?>
569     <select name="H2" size="1">
570         <?php
571         foreach ($SESSION['zusatz'] AS $szusatz) {
572             if ($szusatz[0] == 'H2') {
573                 echo '<option_value="' . $szusatz[4] . '">' . $szusatz[4] . '</option>';
574                 $mmchecked = 1;
575             }
576         }
577         if ($mmchecked != 1) {
578             ?>

```

```

579         <option value="0" selected="selected"></option>
580     <?php
581     }
582     $mmchecked = 0;
583     for ($i = 1; $i <= $petmerk; $i++) {
584         echo '<option value="' . $i . '">' . $i . '</option>';
585     }
586     ?>
587 </select>
588 &nbsp;Haustier/e
589 <?php
590 }
591 ?>
592 <br>
593 <div class="headline">Buchungshinweise und AGB's</div>
594 <?php
595 foreach ($hinweisparam_AS_ $hinwzeile) {
596     /*[1] und [2] sind Start- und Enddatum für den entspr Hinweis*/
597     if (strpos($hinwzeile[0], 'Kautio') {
598         $kautio = strpos($hinwzeile[0], 'EUR');
599         if ($kautio == '') {
600             $kautio = strpos($hinwzeile[0], 'Euro');
601             $kautio = trim(substr($kautio, 4, 10));
602             $kautio = (int) $kautio;
603         }
604         else {
605             $kautio = substr($kautio, 4, 10);
606             $kautio = (int) $kautio;
607         }
608         $_SESSION['zusatz'][1] = array($kautio, 'EUR');
609     }
610
611     if ($hinwzeile[1] == '0' AND $hinwzeile[2] == '0') {
612         echo $hinwzeile[0]. '<br>';
613     }
614     else {
615         echo $hinwzeile[0]. ', gilt nur im Zeitraum vom '.
616         $hinwzeile[1]. ' bis '. $hinwzeile[2]. '!<br>';
617     }
618
619 }
620 ?>
621 <span style="color:#FF0000;" id="fehler13"><?php echo $fehler[12]; ?></span><br>
622 <input type="checkbox" name="buhinweise" value="Y" <?php echo $nobuhinw;?>> Ich bin
623 mit den Buchungshinweisen einverstanden
624 <br>
625 Dieses Objekt geh&#246;rt zum Datenpool von tui-wolters. Deshalb m&#252;ssen sie
626 die AGB's von Tui-Wolters akzeptieren<br>
627 <span style="color:#FF0000;" id="fehler14"><?php echo $fehler[13]; ?></span><br>
628 <input type="checkbox" name="agb" value="Y">Ich akzeptiere die <a
629 href="https://www.tui-wolters.de/agb.html" target="_blank">allgemeinen
630 Gesch&#228;ftsbedingungen</a>
631 <br>
632 <div class="headline">Zahlungsm&#246;glichkeiten</div>
633 <?php
634 if ($_SESSION['payment'][0] == 1 OR $_SESSION['payment'][0] == '') {
635     ?>
636     <input name="zahlungsart" type="radio" value="1"
637     checked="checked">Bankeinzug<br>
638     <?php
639 }
640 else {
641     ?>
642     <input name="zahlungsart" type="radio" value="1" >Bankeinzug<br>
643     <?php
644 }

```

```

645     if ($_SESSION['payment'][0] == 2) {
646         ?>
647         <input name="zahlungsart" type="radio" value="2"
648         checked="checked">Kreditkarte<br>
649         <?php
650     }
651     else {
652         ?>
653         <input name="zahlungsart" type="radio" value="2">Kreditkarte<br>
654         <?php
655     }
656     if ($_SESSION['payment'][0] == 3) {
657         ?>
658         <input name="zahlungsart" type="radio" value="3" <?php echo $disabled; ?>
659         checked="checked">Auf Rechnung (Nur wenn Anreisetag in 6 Wochen oder sp&#228;ter)
660         <?php
661     }
662     else {
663         ?>
664         <input name="zahlungsart" type="radio" value="3" <?php echo $disabled; ?>>Auf
665         Rechnung (Nur wenn Anreisetag in 6 Wochen oder sp&#228;ter)
666         <?php
667     }
668     ?>
669     <br>
670 <?php
671 }
672 /* ***Ende Funktion Wolters Zusatzleistungen*** */
673 ?>
674 <html>
675 <html xmlns="http://www.w3.org/1999/xhtml">
676 <head>
677 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
678 <title>Abfragetests</title>
679 <link href="<?php echo $_downward;?>css/ferienhaus_ibe.css" rel="stylesheet"
680 type="text/css">
681 <script type="text/javascript">
682
683     function validEmail(email) {
684         var strReg = " ^([a-zA-Z0-9_\\.\\-])+(\\@((([a-zA-Z0-9_\\.\\-])+\\.
685 .....)+([a-zA-Z0-9]{2,4})+$)";
686         var regex = new RegExp(strReg);
687         return(regex.test(email));
688     }
689
690     function checkForm1( ){
691         var strFehler = 'N';
692         if (document.personenform.bu_name.value.length<2 ||
693         document.personenform.bu_name.value.indexOf("<") > -1 ||
694         document.personenform.bu_name.value.indexOf(">") > -1) {
695             document.all.fehler2.innerHTML = 'Name_angeben';
696             document.personenform.bu_name.focus();
697             strFehler = 'Y';
698         }
699         else {
700             document.all.fehler2.innerHTML = '';
701         }
702
703         if (document.personenform.bu_vorname.value.length<2 ||
704         document.personenform.bu_vorname.value.indexOf("<") > -1 ||
705         document.personenform.bu_vorname.value.indexOf(">") > -1) {
706             document.all.fehler3.innerHTML = 'Vorname_angeben';
707             document.personenform.bu_vorname.focus();
708             strFehler = 'Y';
709         }
710         else {

```

```

711     document.all.fehler3.innerHTML = '';
712 }
713
714 if (document.personenform.bu_strasse.value.length<5 ||
715 document.personenform.bu_strasse.value.indexOf("<") > -1 ||
716 document.personenform.bu_strasse.value.indexOf(">") > -1) {
717     document.all.fehler4.innerHTML = 'Strassennamen_angeben';
718     document.personenform.bu_strasse.focus();
719     strFehler = 'Y';
720 }
721 else {
722     document.all.fehler4.innerHTML = '';
723 }
724
725 if (document.personenform.bu_plz.value.length > 10 ||
726 document.personenform.bu_plz.value.length<5) {
727     document.all.fehler5.innerHTML = 'Bitte_Postleitzahl_angeben';
728     document.personenform.bu_plz.focus();
729     strFehler = 'Y';
730 }
731 else {
732     document.all.fehler5.innerHTML = '';
733 }
734 if (document.personenform.bu_ort.value.length<2 ||
735 document.personenform.bu_ort.value.indexOf("<") > -1 ||
736 document.personenform.bu_ort.value.indexOf(">") > -1) {
737     document.all.fehler6.innerHTML = 'Ortsname_fehlt';
738     document.personenform.bu_ort.focus();
739     strFehler = 'Y';
740 }
741 else {
742     document.all.fehler6.innerHTML = '';
743 }
744
745 if (document.personenform.bu_land.value == 'kA' ) {
746     document.all.fehler7.innerHTML = 'Kein_Land_angegeben';
747     document.personenform.bu_land.focus();
748     strFehler = 'Y';
749 }
750 else {
751     document.all.fehler7.innerHTML = '';
752 }
753 if (document.personenform.bu_telefon.value.match(/\D/) ||
754 document.personenform.bu_telefon.value.length<8) {
755     document.all.fehler8.innerHTML = 'Telefonnummer_vergessen';
756     document.personenform.bu_telefon.focus();
757     strFehler = 'Y';
758 }
759 else {
760     document.all.fehler8.innerHTML = '';
761 }
762 if (!validEmail(document.personenform.bu_mail.value)) {
763     document.all.fehler10.innerHTML = 'Keine_korrekte_E-Mail_Adresse
764 .....angegeben';
765     document.personenform.bu_mail.focus();
766     strFehler = 'Y';
767 }
768 else {
769     document.all.fehler10.innerHTML = '';
770 }
771 <?php
772 if ($anbieter == 1) {
773 ?>
774     if (!document.personenform.agb.checked) {
775         document.all.agberror.innerHTML = 'Bitte_die_AGBs_akzeptieren';
776         document.personenform.agb.focus();

```

```

777         strFehler = 'Y';
778     }
779     else {
780         document.all.fehler14.innerHTML = '';
781     }
782 <?php
783 }
784 if ($anbieter == 2) {
785     ?>
786     /* buhinweise agb */
787     if (!document.personenform.buhinweise.checked) {
788         document.all.fehler13.innerHTML = 'Bitte die Buchungshinweise
789 ..... akzeptieren';
790         document.personenform.buhinweise.focus();
791         strFehler = 'Y';
792     }
793     else {
794         document.all.fehler13.innerHTML = '';
795     }
796
797     if (!document.personenform.agb.checked) {
798         document.all.fehler14.innerHTML = 'Bitte die AGBs akzeptieren';
799         document.personenform.agb.focus();
800         strFehler = 'Y';
801     }
802     else {
803         document.all.fehler14.innerHTML = '';
804     }
805 <?php
806 }
807 ?>
808 if (strFehler == 'Y') {
809     return(false);
810 }
811 }
812 </script>
813
814 <script type="text/javascript">
815
816
817 </script>
818
819 </head>
820 <body>
821 <!-- ***Hauptcontainer Start*** -->
822 <div id="main">
823     <div class="bu_header"> <span class="red">Adressdaten</span>&nbsp; ||
824     &nbsp; weitere Optionen&nbsp; ||&nbsp; Angaben pruefen&nbsp; ||
825     &nbsp; Buchung best&#228;tigen</div>
826 <?php
827 require_once '../header.php';
828 ?>
829 <div class="headline">Adressdaten</div>
830
831 <form name="personenform" action="buchen2.php" onsubmit="return _checkForm1();"
832 method="POST">
833 <!-- ***Start Erfassung Personendaten*** -->
834 <div class="bu_adressdaten">
835     <table align="center">
836     <!-- <form name="personenform" onsubmit="return _checkForm1();"
837     action="wolters_buchung2.php?oref=<?php echo _$objektreferenz; _?>&okat=<?php
838 ..... echo _$objektkategorie; _?>&date=<?php echo _$aufenthalt_start; _?>&days=<?php
839 ..... echo _$aufenthaltsdauer; _?>&anbieter=<?php echo _$anbieter; _?>" method="post">
840     // -->
841     <!-- <form name="personenform" onsubmit="return _checkForm();"
842     action="buchungsbuchungswolters.php?oref=<?php echo _$objektreferenz;

```

```

843 .....?>&okat=<?php_echo_$objektkategorie;_?>&date=<?php_echo_$aufenthalt_start;
844 .....?>&days=<?php_echo_$aufenthaltsdauer;_?>&anbieter=<?php_echo_$anbieter;_?>"
845 method="post"> //--->
846     <tr>
847         <td>&nbsp;</td>
848         <td id="fehler1" style="color:#FF0000;">&nbsp;<?php_echo $fehler[0];
849         ?></td>
850     </tr>
851     <tr>
852         <td>Anrede:</td>
853         <td>
854             <?php
855             if ($_SESSION['person'][0] == 2) {
856                 ?>
857                 <input type="radio" name="bu_anrede" value="1">Herr</input>
858                 <input type="radio" name="bu_anrede" value="2"
859                 checked="checked">Frau</input>
860             <?php
861             }
862             else {
863                 ?>
864                 <input type="radio" name="bu_anrede" value="1"
865                 checked="checked">Herr</input>
866                 <input type="radio" name="bu_anrede" value="2">Frau</input>
867             <?php
868             }
869             ?>
870         </td>
871     </tr>
872     <tr>
873         <td>&nbsp;</td>
874         <td id="fehler2" style="color:#FF0000;">&nbsp;<?php_echo $fehler[1];
875         ?></td>
876     </tr>
877     <tr>
878         <td>Name:</td>
879         <td>
880             <input name="bu_name" type="text" value="<?php_echo
881 .....$_SESSION['person'][1];_?>" maxlength="30" size="30"></input>
882             </td>
883     </tr>
884     <tr>
885         <td>&nbsp;</td>
886         <td id="fehler3" style="color:#FF0000;">&nbsp;<?php_echo $fehler[2];
887         ?></td>
888     </tr>
889     <tr>
890         <td>Vorname:</td>
891         <td>
892             <input name="bu_vorname" type="text" value="<?php_echo
893 .....$_SESSION['person'][2];_?>" maxlength="30" size="30"></input>
894             </td>
895     </tr>
896     <tr>
897         <td>&nbsp;</td>
898         <td id="fehler4" style="color:#FF0000;">&nbsp;<?php_echo $fehler[3];
899         ?></td>
900     </tr>
901     <tr>
902         <td>Strasse:</td>
903         <td>
904             <input name="bu_strasse" type="text" value="<?php_echo
905 .....$_SESSION['person'][3];_?>" maxlength="30" size="30"></input>
906             </td>
907     </tr>
908     <tr>

```



```

909         <td>&nbsp;</td>
910         <td id="fehler5" style="color:#FF0000;">&nbsp;<?php echo $fehler[4];
911         ?></td>
912     </tr>
913     <tr>
914         <td>Postleitzahl:*</td>
915         <td>
916             <input name="bu_plz" type="text" value="<?php echo
917 .....$_SESSION['person'][4];_?>" maxlength="10" size="10"></input>
918         </td>
919     </tr>
920     <tr>
921         <td>&nbsp;</td>
922         <td id="fehler6" style="color:#FF0000;">&nbsp;<?php echo $fehler[5];
923         ?></td>
924     </tr>
925     <tr>
926         <td>Ort:*</td>
927         <td>
928             <input name="bu_ort" type="text" value="<?php echo
929 .....$_SESSION['person'][5];_?>" maxlength="30" size="30"></input>
930         </td>
931     </tr>
932     <tr>
933         <td>&nbsp;</td>
934         <td id="fehler7" style="color:#FF0000;">&nbsp;<?php echo $fehler[6];
935         ?></td>
936     </tr>
937     <tr>
938         <td>Land:*</td>
939         <td>
940             <select name="bu_land" size="1">
941                 <?php
942                 $primary = 0;
943                 if (isset($_SESSION['person'][6]) AND $_SESSION['person'][6] != '' )
944                 {
945                     $primary = 1;
946                     $idland = $_SESSION['person'][6];
947                     if ($idland != '') {
948                         $query = "SELECT _Land FROM _ref_land WHERE _IDLand=_ '$idland'
949 .....LIMIT 1";
950                         $result = mysql_query($query);
951                         while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
952                             $land = $line['Land'];
953                         }
954                     }
955                 }
956                 /* else {
957                 } */
958                 if ($primary == 1) {
959                     ?>
960                     <option value="kA">Auswahl treffen </option>
961                 <?php
962                 }
963                 else {
964                     ?>
965                     <option value="kA" selected="selected">Auswahl treffen </option>
966                 <?php
967                 }
968                 $landquery = "SELECT _IDLand, _Land FROM $tabelle . menu_land ORDER BY
969 .....id";
970                 $landreq = mysql_query($landquery);
971                 while($land_line = mysql_fetch_array($landreq, MYSQL_ASSOC)) {
972                     if ($primary == 1 AND utf8_encode($land_line['IDLand']) ==
973                     $idland) {
974                         echo '<option value="'.utf8_encode($land_line['IDLand']).

```

```

975         '","
976 .....selected="selected">' . utf8_encode( $land_line[ 'Land' ] ). '</option>';
977     }
978     else {
979         echo ' <option_value=" ' . utf8_encode( $land_line[ 'IDLand' ] ).
980         '">' . utf8_encode( $land_line[ 'Land' ] ). '</option>';
981     }
982 }
983 $primary = 0;
984 ?>
985 </select>
986 </td>
987 </tr>
988 <tr>
989     <td>&nbsp;</td>
990     <td id=" fehler8" style=" color:#FF0000;">&nbsp;<?php echo
991     $fehler[7]; ?></td>
992 </tr>
993 <tr>
994     <td>Telefon Tags&#252;ber:*</td>
995     <td>
996         <!-- <input type="text" name="quntity" class="integer"/> -->
997         <input name="bu_telefon" type="text" value="<?php echo
998 .....$_SESSION[ 'person ' ][7]; ?>" maxlength="30" size="30"></input>
999     </td>
1000 </tr>
1001 <tr>
1002     <td>Fax<br>Telefon Abends:</td>
1003     <td>
1004         <input name="bu_telefon2" type="text" value="<?php echo
1005 .....$_SESSION[ 'person ' ][8]; ?>" maxlength="30" size="30"></input>
1006     </td>
1007 </tr>
1008 <?php if ( $anbieter == '2' ) { ?>
1009 <tr>
1010     <td>Mobil Tel:</td>
1011     <td>
1012         <input name="bu_telefon3" type="text" value="<?php echo
1013 .....$_SESSION[ 'person ' ][9]; ?>" maxlength="30" size="30"></input>
1014     </td>
1015 </tr>
1016 <?php } ?>
1017 <tr>
1018     <td>&nbsp;</td>
1019     <td id=" fehler9" style=" color:#FF0000;">&nbsp;</td>
1020 </tr>
1021 <tr>
1022     <td>&nbsp;</td>
1023     <td id=" fehler10" style=" color:#FF0000;">&nbsp;<?php echo $fehler[8];
1024     ?></td>
1025 </tr>
1026 <tr>
1027     <td>E-Mail:*</td>
1028     <td>
1029         <input name="bu_mail" type="text" value="<?php echo
1030 .....$_SESSION[ 'person ' ][10]; ?>" maxlength="30" size="30"></input>
1031     </td>
1032 </tr>
1033 <tr>
1034     <td>&nbsp;</td>
1035     <td id=" fehler11" style=" color:#FF0000;">&nbsp;<?php echo $fehler[9];
1036     ?></td>
1037 </tr>
1038 <tr>
1039     <td>Reisende:*</td>
1040     <td>

```

```

1041         <select name="bu_personen" size="1">
1042         <?php
1043         $primary = 0;
1044         if (isset($_SESSION['person'][11]) AND $_SESSION['person'][11] != '') {
1045             $primary = 1;
1046         }
1047         for ($i = 1; $i <= $anz_personen; $i++) {
1048             if ($primary == 1 AND $i == $_SESSION['person'][11]) {
1049                 echo '<option value="' . $i . '" selected="selected">' . $i . '</option>';
1050             }
1051             elseif ($primary == 0 AND $i == 1) {
1052                 echo '<option value="' . $i . '" selected="selected">' . $i . '</option>';
1053             }
1054             else {
1055                 echo '<option value="' . $i . '">' . $i . '</option>';
1056             }
1057         }
1058         ?>
1059     </select>
1060 </td>
1061 </tr>
1062 <tr>
1063     <td>davon Kinder (bis 14J., je nach Anbieter):</td>
1064     <td>
1065         <select name="bu_kinder" size="1">
1066         <?php
1067         $primary = 0;
1068         if (isset($_SESSION['person'][12]) AND $_SESSION['person'][12] != '') {
1069             $primary = 1;
1070         }
1071
1072         if ($anbieter == '2') {
1073             if ($primary == 0) {
1074                 echo '<option value="0" selected="selected">Keine</option>';
1075             }
1076             else {
1077                 echo '<option value="0">Keine</option>';
1078             }
1079             for ($i = 1; $i <= $anz_kinder; $i++) {
1080                 if ($primary == 1 AND $i == $_SESSION['person'][12]) {
1081                     echo '<option value="' . $i . '"
1082 .....selected="selected">' . $i . '</option>';
1083                 }
1084                 else {
1085                     echo '<option value="' . $i . '">' . $i . '</option>';
1086                 }
1087             }
1088         }
1089         elseif($anbieter == '1') {
1090             if ($primary == 0) {
1091                 echo '<option value="0" selected="selected">Keine</option>';
1092             }
1093             else {
1094                 echo '<option value="0">Keine</option>';
1095             }
1096
1097             for ($i = 1; $i <= $anz_personen - 1; $i++) {
1098                 if ($primary == 1 AND $i == $_SESSION['person'][12]) {
1099                     echo '<option value="' . $i . '"
1100 .....selected="selected">' . $i . '</option>';
1101                 }
1102                 else {
1103                     echo '<option value="' . $i . '">' . $i . '</option>';
1104                 }
1105             }
1106         }

```

```

1107         ?>
1108     </select>
1109 </td>
1110 </tr>
1111 <?php //} ?>
1112 <?php if ($anbieter == '2') { ?>
1113 <tr>
1114     <td>Reiser&#252;cktritts-<br>kostenversicherung:</td>
1115     <td>
1116         <?php
1117             if ($.SESSION['zusatz'][0] == 'Y') {
1118                 echo '<input type="checkbox" name="rrkv" value="Y"
1119 .....checked="checked"></input>';
1120             }
1121             else {
1122                 ?>
1123                 <input type="checkbox" name="rrkv" value="Y"></input>
1124             <?php
1125             }
1126             ?>
1127         </td>
1128     </tr>
1129 <?php } ?>
1130 <tr>
1131     <td>&nbsp;</td>
1132     <td id="fehler12" style="color:#FF0000;">&nbsp;<?php echo $fehler[10];
1133     ?></td>
1134 </tr>
1135 <tr>
1136     <td>Zus. Angaben</td>
1137     <td>
1138         <textarea name="bu_zusatztext" cols="30" rows="5" name="zusatztext"
1139         style="white-space:pre;"><?php echo $.SESSION['person'][13];
1140         ?></textarea>
1141     </td>
1142 </tr>
1143 </table>
1144 </div>
1145 <!-- ***Ende Erfassung Personendaten*** //-->
1146
1147 <div class="headline">Zusatzleistungen</div>
1148
1149 <div>
1150     <?php
1151     if ($anbieter == '1') {
1152         call_user_func('zusatzleistungen_interhome',$pfad, $anbieter,
1153         $objektreferenz, $objektkategorie, $anreisedatum, $aufenthaltsdauer,
1154         $fehler);
1155     }
1156     elseif ($anbieter == '2') {
1157         call_user_func('zusatzleistungen_wolters',$pfad, $anbieter,
1158         $objektreferenz, $objektkategorie, $anreisedatum, $aufenthaltsdauer,
1159         $fehler);
1160     }
1161     ?>
1162 </div>
1163 <div>
1164     <input type="hidden" name="checkup" value="1">
1165     <input type="hidden" name="sid" value="<?php echo $_$sessionid;_?>">
1166     <input type="submit" value="Absenden">
1167 </div>
1168 </form>
1169 </div>
1170 <!-- ***Hauptcontainer Ende*** //-->
1171 </body>
1172 </html>

```



```
62     document.zahlungsart1.kto.value.length<8) {
63         document.all.fehler13.innerHTML = 'Kontonummer_angeben';
64         document.zahlungsart1.kto.focus();
65         strFehler = 'Y';
66     }
67     else {
68         document.all.fehler13.innerHTML = '';
69     }
70
71     if (document.zahlungsart1.blz.value.match(/\D/) ||
72     document.zahlungsart1.blz.value.length<8) {
73         document.all.fehler14.innerHTML = 'Bankleitzahl_angeben';
74         document.zahlungsart1.blz.focus();
75         strFehler = 'Y';
76     }
77     else {
78         document.all.fehler14.innerHTML = '';
79     }
80
81     if (document.zahlungsart1.bank.value.length<8 ||
82     document.zahlungsart1.bank.value.indexOf("<") > -1 ||
83     document.zahlungsart1.bank.value.indexOf(">") > -1) {
84         document.all.fehler15.innerHTML = 'Kreditinstitut_angeben';
85         document.zahlungsart1.bank.focus();
86         strFehler = 'Y';
87     }
88     else {
89         document.all.fehler15.innerHTML = '';
90     }
91
92     if (strFehler == 'Y') {
93         return(false);
94     }
95 }
96
97 function checkForm3( ){
98     var strFehler = 'N';
99     var Datum = new Date();
100     var Jahr = Datum.getFullYear();
101     var Monat = Datum.getMonth();
102
103     if (document.zahlungsart2.cctype.value == '0') {
104         document.all.fehler16.innerHTML = 'Kreditkartentyp_ausw&#228;hlen';
105         document.zahlungsart2.cctype.focus();
106         strFehler = 'Y';
107     }
108     else {
109         document.all.fehler16.innerHTML = '';
110     }
111
112     if (document.zahlungsart2.ccnum.value.match(/\D/) ||
113     document.zahlungsart2.ccnum.value.length < 16) {
114         document.all.ccnumerror.innerHTML = 'Kreditkartennummer_angeben';
115         document.zahlungsart2.ccnum.focus();
116         strFehler = 'Y';
117     }
118     else {
119         document.all.ccnumerror.innerHTML = '';
120     }
121
122     if (document.zahlungsart2.ccexpirem.value <= Monat &&
123     document.zahlungsart2.ccexpirey.value <= Jahr) {
124         document.all.ccexpireerror.innerHTML = 'Kreditkarte_abgelaufen';
125         document.zahlungsart2.ccexpirem.focus();
126         strFehler = 'Y';
127     }
```

```

128     else {
129         document.all.ccexpireerror.innerHTML = '';
130     }
131
132     if (document.zahlungsart2.ccfname.value.length < 3 ||
133     document.zahlungsart2.ccfname.value.indexOf("<") > -1 ||
134     document.zahlungsart2.ccfname.value.indexOf(">") > -1) {
135         document.all.ccfnameerror.innerHTML = 'Vornamen_angeben';
136         document.zahlungsart2.ccfname.focus();
137         strFehler = 'Y';
138     }
139     else {
140         document.all.ccfnameerror.innerHTML = '';
141     }
142
143     if (document.zahlungsart2.cclname.value.length < 3 ||
144     document.zahlungsart2.cclname.value.indexOf("<") > -1 ||
145     document.zahlungsart2.cclname.value.indexOf(">") > -1) {
146         document.all.cclnameerror.innerHTML = 'Nachnamen_angeben';
147         document.zahlungsart2.cclname.focus();
148         strFehler = 'Y';
149     }
150     else {
151         document.all.cclnameerror.innerHTML = '';
152     }
153
154     if (strFehler == 'Y') {
155         return(false);
156     }
157 }
158
159 function checkForm4( ){
160     var strFehler = 'N';
161     if (!document.zahlungsart3.rechnung.checked) {
162         document.all.fehler21.innerHTML = 'Bitte_Buchung_auf_Rechnung_bestätigen';
163         document.zahlungsart3.rechnung.focus();
164         strFehler = 'Y';
165     }
166     else {
167         document.all.fehler21.innerHTML = '';
168     }
169
170     if (strFehler == 'Y') {
171         return(false);
172     }
173 }
174 </script>
175 </head>
176 <body>
177 <!-- ***Hauptcontainer Start*** //—>
178 <div id="main">
179     <div class="bu_adressdaten"> <span>Adressdaten</span>&nbsp;&nbsp;&||&nbsp;&nbsp;&<span
180     class="red">Zahlungsangaben</span>&nbsp;&nbsp;&||&nbsp;&nbsp;&Angaben
181     pruefen&nbsp;&||&nbsp;&nbsp;&Buchung best&#228;tigen</div>
182     <?php
183     require_once '../header.php';
184     ?>
185     <div class="headline">Zahlungsangaben</div>
186     <?php
187     $zahlungstyp = $_SESSION['payment'][0];
188
189     if ($zahlungstyp == 1) {
190     ?>
191         <h2>Zahlung per Bankeinzug</h2>
192         <div class="bu_pay">
193             <form name="zahlungsart1" onsubmit="return_checkForm2();"

```



```

260      <?php
261      $primary = 0;
262      $cctype = $_SESSION[ 'payment' ][2][0];
263      $query = "SELECT_*_FROM_mon_creditcard WHERE_kurz_=_ '$cctype'
264      .....LIMIT_1";
265      $result = mysql_query($query);
266      if (mysql_num_rows($result) < 1) {
267          $primary = 0;
268      }
269      else {
270          /*Zusweisung Kreditkartennamen zum Kürzel*/
271          while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
272              $card = $line[ 'typ' ];
273          }
274          $primary = 1;
275      }
276      if ($anbieter == 1) {
277          if ($primary == 1) {
278              echo '<option_value="'. $cctype. '"
279      .....selected="selected">' . $card. '</option>';
280              echo '<option_value="MAST">Mastercard</option>';
281              echo '<option_value="VISA">Visacard</option>';
282          }
283          else {
284              echo '<option_value="MAST"
285      .....selected="selected">Mastercard</option>';
286              echo '<option_value="VISA">Visacard</option>';
287          }
288          $primary = 0;
289      }
290      elseif ($anbieter == 2) {
291          if ($primary == 1) {
292              ?>
293              <option value="<?php_echo_$cctype;?>"
294              selected="selected"><?php_echo $card;?></option>
295              <option value="AX">American Express</option>
296              <option value="MC">Mastercard</option>
297              <option value="VI">Visa-Card</option>
298              <option value="TU">TUI-Card</option>
299              <?php
300          }
301          else {
302              ?>
303              <option selected value="AX">American Express</option>
304              <option value="MC">Mastercard</option>
305              <option value="VI">Visa-Card</option>
306              <option value="TU">TUI-Card</option>
307              <?php
308          }
309          $primary = 0;
310      }
311      ?>
312
313      </select>
314      </td>
315      </tr>
316      <tr>
317          <td>&nbsp;</td>
318          <td style="color:#FF0000;" id="ccnumerror"><?php_echo $fehler[0];
319          ?></td>
320      </tr>
321      <tr>
322          <td>
323              Kartennummer:
324          </td>
325          <td>

```

```

326         <input name="ccnum" type="text" maxlength="16" size="16"
327         value="<?php_echo_$_SESSION['payment'][2][1];_?>"></input>
328     </td>
329 </tr>
330 <tr>
331     <td>&nbsp;</td>
332     <td style="color:#FF0000;" id="ccvererror"><?php_echo $fehler[1];
333     ?></td>
334 </tr>
335 <tr>
336     <td>
337         Verifizierungsnummer:
338     </td>
339     <td>
340         <input name="ccver" type="text" maxlength="3" size="3" value="<?php
341         .....echo_$_SESSION['payment'][2][2];_?>"></input>
342     </td>
343 </tr>
344 <tr>
345     <td>&nbsp;</td>
346     <td style="color:#FF0000;" id="ccexpireerror"><?php_echo $fehler[2];
347     ?></td>
348 </tr>
349 <tr>
350     <td>
351         Verfallsdatum:
352     </td>
353     <td>
354         <select name="ccexpire">
355             <?php
356                 $thisyear = date("Y");
357                 $ccyear = $_SESSION['payment'][2][3];
358                 $ccmonth = $_SESSION['payment'][2][4];
359                 $selector = NULL;
360                 $primary = 0;
361                 if (isset($ccyear) AND $ccyear != '') {
362                     ?>
363                     <option value="<?php_echo_$_ccyear;_?>" selected="selected"><?php
364                     .....echo $ccyear; ?></option>
365                     <?php
366                 }
367                 else {
368                     ?>
369                     <option value="<?php_echo_$_thisyear;_?>"
370                     .....selected="selected"><?php_echo $thisyear; ?></option>
371                     <?php
372                 }
373                 for ($i = 1; $i <= 15; $i++) {
374                     $forwardyear = $thisyear+$i;
375                     .....echo '<option
376                     .....value="'. $forwardyear .'">'. $forwardyear . '</option>';
377                     }
378                     ?>
379                 </select>
380                 <select name="ccexpirem">
381                     <?php
382                         $thismonth = date("n");
383                         if (isset($ccmonth) AND $ccmonth!= '') {
384                             $primary = 1;
385                         }
386                         for ($i = 1; $i <= 12; $i++) {
387                             if ($primary == 1 AND $i == $ccmonth) {
388                                 .....echo '<option_value="'. $i . '
389                                 .....selected="selected">'. $i . '</option>';
390                             }
391                             elseif ($primary == 0 AND $i == $thismonth) {

```

```

392         echo '<option_value="' . $i . '
393 .....selected="selected"'>' . $i . '</option>';
394     }
395     else {
396         echo '<option_value="' . $i . ' . $selector . "'>' . $i . '</option>';
397     }
398 }
399 ?>
400 </select>
401 </td>
402 </tr>
403 <tr>
404     <td colspan="2">
405         Kreditkarteninhaber
406     </td>
407 </tr>
408 <tr>
409     <td>&nbsp;</td>
410     <td style="color:#FF0000;" id="cclnameerror"><?php echo $fehler[3];
411     ?></td>
412 </tr>
413 <tr>
414     <td>
415         Name:
416     </td>
417     <td>
418         <input name="cclname" type="text" value="<?php echo
419 .....$_SESSION['payment'][2][6];? >"></input>
420     </td>
421 </tr>
422 <tr>
423     <td>&nbsp;</td>
424     <td style="color:#FF0000;" id="ccfnameerror"><?php echo $fehler[4];
425     ?></td>
426 </tr>
427 <tr>
428     <td>
429         Vorname:
430     </td>
431     <td>
432         <input name="ccfname" type="text" value="<?php echo
433 .....$_SESSION['payment'][2][5];? >"></input>
434     </td>
435 </tr>
436 </table>
437 <input type="hidden" name="checkup" value="3"></input>
438 <input type="hidden" name="sid" value="<?php echo $_SESSION['sid'];? >">
439 <input type="submit" value="weiter">
440 </form>
441 </div>
442 <?php
443 }
444 elseif($zahlungstyp == 3) {
445     ?>
446     <h2>Zahlung auf Rechnung</h2>
447     <div class="bu_pay">
448         <form name="zahlungsart3" onsubmit="return checkForm4();"
449         action="buchen3.php" method="post">
450             <table>
451                 <tr>
452                     <td id="fehler21" style="color:#FF0000;">&nbsp;<?php echo $fehler[0];
453                     ?></td>
454                 </tr>
455             </table>
456             <input type="checkbox" name="rechnung" value="Y">Gegen Rechnung
457             <?php

```

```

458         if ($anbieter == 2) {
459             echo '(nur hier f&#228;llt ein Bearbeitungs-Entgelt in H&#246;he von
460 .....5.00 EUR an)<br>';
461         }
462         else {
463             echo '<br>';
464         }
465     ?>
466
467     <input type="hidden" name="checkup" value="4"></input>
468     <input type="hidden" name="sid" value="<?php echo $sessionid; ?>">
469     <input type="submit" value="weiter">
470     <!-- ***Ende Form Zahlungsart Rechnung*** /-->
471 </form>
472 </div>
473 <?php
474 }
475 else {
476     echo 'Warnung, Fehler bei Erfassen des Zahlungstypes. Buchug kann nicht
477 .....fortgesetzt werden.<br>Bitte einen Administrator kontaktieren!';
478 }
479 ?>
480
481 </div>
482 <!-- ***Hauptcontainer Ende*** /-->
483 </body>
484 </html>

```

G.13. Buchungsvorgang Teil 3

Listing G.11: Quelltext Buchen 3

```

1 <?php
2
3 /**
4  * Buchungsbereich 3, Ausgabe und Übersicht der Daten
5  * Ausgabe eines Headers mit Angaben über Objektpreis,
6  * Preis aller Zusatzleistungen. Darstellung der
7  * personenbezogenen Daten, Übersicht über Zusatzleistungen
8  * und Berechnung der Preise sowie Umrechnung von einer
9  * Landeswährung in Euro. Angaben über Reisezeitraum und
10 * Anreisetag und Möglichkeiten die Daten zu korrigieren
11 * oder abzuschicken
12 */
13
14 $sessionid = $_REQUEST['sid'];
15 if (!isset($_COOKIE['PHPSESSID'])) {
16     session_id($sessionid);
17 }
18 session_start();
19 echo session_id(). '—<br>';
20 echo $sessionid;
21
22 require_once 'buchen_varcheck.php';
23
24 /* Abfrageschnittstellen einbinden */
25 $pfad = getenv("DOCUMENT_ROOT");
26 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
27 include $datenbankzugriff;
28 $inc_objektklasse = $pfad.'/include/class_objekte_db.php';
29 include $inc_objektklasse;
30 $inc_geldklasse = $pfad.'/include/class_geld.php';
31 include $inc_geldklasse;
32

```

```
33 /* Abfrageschnittstellen initialisieren */
34 $objektdaten = new datensaetze;
35 $moneur = new changeto_eur;
36
37 /* Objektdaten referenzieren */
38 $objektreferenz = $_SESSION['objekt'][0];
39 $objektkategorie = $_SESSION['objekt'][1];
40 $anreisedatum = $_SESSION['objekt'][2];
41 $aufenthaltsdauer = $_SESSION['objekt'][3];
42 $anbieter = $_SESSION['objekt'][4];
43
44 /* Variablen deklarieren */
45 $stabelle='usr_web31_1';
46 $fehler = $_SESSION['meldung'];
47
48 function zusatzleistungen_interhome ($pfad, $anbieter, $objektreferenz,
49 $objektkategorie, $anreisedatum, $aufenthaltsdauer, $fehler, $sessionid) {
50     $objektpreis = $_SESSION['objekt'][5];
51     $versicherung = $_SESSION['zusatz'][0];
52     $objekt_gesamt = $objektpreis[0]+$versicherung[2];
53     $objektdaten = new datensaetze;
54     $moneur = new changeto_eur;
55     $preiseur = 0;
56     $preiseurges = 0;
57     ?>
58     <div class="bu_zubview">
59         <table>
60             <tr>
61                 <td>
62                     Preis des Objektes:
63                 </td>
64                 <td>
65                     <?php
66                     if ($objektpreis[1] == 'EUR') {
67                         echo $objektpreis[0]. '&nbsp;'. $objektpreis[1];
68                     }
69                     else {
70                         $moneur->eurpreis($objektpreis[1], $objektpreis[0]);
71                         $europreis = $moneur->preis_eur;
72                         echo $objektpreis[0]. '&nbsp;'. $objektpreis[1].
73                             '&nbsp;/&nbsp;'. $europreis. '&nbsp;EUR';
74                         $preiseurges = $preiseurges+$preiseur;
75                     }
76                     ?>
77                 </td>
78             </tr>
79             <tr>
80                 <td>
81                     Versicherung:
82                 </td>
83                 <td>
84                     <?php
85                     if ($versicherung[3] == 'EUR') {
86                         echo $versicherung[2]. '&nbsp;'. $versicherung[3];
87                     }
88                     else {
89                         $moneur->eurpreis($objektpreis[1], $versicherung[2]);
90                         $europreis = $moneur->preis_eur;
91                         echo $versicherung[2]. '&nbsp;'. $versicherung[3].
92                             '&nbsp;/&nbsp;'. $europreis. '&nbsp;EUR';
93                         $preiseurges = $preiseurges+$preiseur;
94                     }
95                     ?>
96                 </td>
97             </tr>
98             <tr>
```

```

99         <td>
100             <span class="fett">Objekt Gesamt:</span>
101         </td>
102         <td>
103             <span class="fett">
104                 <?php
105                     if ($objektpreis[1] == 'EUR') {
106                         echo $objekt_gesamt.'&nbsp;'.$objektpreis[1];
107                     }
108                     else {
109                         echo $objekt_gesamt.'&nbsp;'.$objektpreis[1].
110                             '&nbsp;/'.$preiseurges.'&nbsp;EUR';
111                     }
112                     $waehrungobj = $objektpreis[1];
113                 ?>
114             </span>
115         </td>
116     </tr>
117     <tr>
118         <td>
119             &nbsp;
120         </td>
121         <td>
122             &nbsp;
123         </td>
124     </tr>
125     <tr>
126         <td>
127             <span class="kursiv">Verpflichtende Zusatzleistungen:</span>
128         </td>
129         <td>
130             <!-- <span class="fett"><?php echo
131                 $_SESSION['zusatz'][1][0]. '&nbsp;'.$_SESSION['zusatz'][1][1];? >
132             </span> //-->&nbsp;
133         </td>
134     </tr>
135     <?php
136     $waehrung_alternativ = '';
137     $preiseur = 0;
138     $preiseurges = 0;
139     $zusatz_berechnung = array ('pro_Tag', 'pro-Woche', 'pro-Person',
140     'pro-Person_Tag', 'pro-Person-Woche');
141     $zusatz = $_SESSION['zusatz'];
142     $gesamtkosten = 0;
143     for ($i=31; $i < 42; $i++) {
144         if (is_array($zusatz[$i])) {
145             $waehrung_alternativ = trim($zusatz[$i][3]);
146             $zusatz_leistungsart = trim($zusatz[$i][1]);
147             $zusatz_berechnung1 = trim($zusatz[$i][5]);
148             $zusatz_berechnung1 = str_replace("/", "-", $zusatz_berechnung1);
149             $zusatz_berechnung1 = str_replace("_", "-", $zusatz_berechnung1);
150             ?>
151         <tr>
152             <td>
153                 <?php
154                 /* Beschreibung der Leistung */
155                 echo utf8_decode($zusatz[$i][1]). '&nbsp;';
156                 /* Anzahl und Preis*/
157                 if ($zusatz[$i][3] == 'EUR') {
158                     echo $zusatz[$i][1]. '&nbsp;'.$zusatz[$i][2].
159                         '&nbsp;'.$zusatz[$i][3]. '<br>';
160                 }
161                 else {
162                     $moneur->eurpreis($zusatz[$i][3], $zusatz[$i][2]);
163                     $europreis = $moneur->preis_eur;
164                     echo $zusatz[$i][4]. 'x, &nbsp;'.$zusatz[$i][5]. '<br>&nbsp;'.

```

```

165      $zusatz[$i][2]. '&nbsp;'. $zusatz[$i][3].
166      '&nbsp;/&nbsp;'. $europreis. '&nbsp;EUR';
167
168      }
169      ?>
170 </td>
171 <td>
172 <?php
173     /* Berechnung Pro Woche */
174     if (preg_match("/\b$zusatz_berechnung[1]\b/i", $zusatz_berechnung1)) {
175         $preis[$i] = $aufenthaltsdauer/7*$zusatz[$i][2];
176     }
177     /* Berechnung Pro Tag */
178     elseif (preg_match("/\b$zusatz_berechnung[0]\b/i",
179 $zusatz_berechnung1)) {
180         $preis[$i] = $aufenthaltsdauer*$zusatz[$i][2];
181     }
182     /* Berechnung Pro Person */
183     elseif (preg_match("/\b$zusatz_berechnung[2]\b/i",
184 $zusatz_berechnung1)) {
185         if ($zusatz_leistungsart == 'Babyausstattung' OR
186 $zusatz_leistungsart == 'Babybett' OR $zusatz_leistungsart ==
187 'Highchair' OR $zusatz_leistungsart == 'Hochstuhl' OR
188 $zusatz_leistungsart == 'Kinderbett') {
189             $preis[$i] = $_SESSION['person'][12]*$zusatz[$i][2];
190         }
191         else {
192             $preis[$i] = $_SESSION['person'][11]*$zusatz[$i][2];
193         }
194     }
195     /* Berechnung Pro Person-Tag */
196     elseif (preg_match("/\b$zusatz_berechnung[3]\b/i",
197 $zusatz_berechnung1)) {
198         if ($zusatz_leistungsart == 'Babyausstattung' OR
199 $zusatz_leistungsart == 'Babybett' OR $zusatz_leistungsart ==
200 'Highchair' OR $zusatz_leistungsart == 'Hochstuhl' OR
201 $zusatz_leistungsart == 'Kinderbett') {
202             $preis[$i] =
203             $aufenthaltsdauer*$_SESSION['person'][12]*$zusatz[$i][2];
204         }
205         else {
206             $preis[$i] =
207             $aufenthaltsdauer*$_SESSION['person'][11]*$zusatz[$i][2];
208         }
209     }
210     /* Berechnung Pro Person-Woche */
211     elseif (preg_match("/\b$zusatz_berechnung[4]\b/i",
212 $zusatz_berechnung1)) {
213         if ($zusatz_leistungsart == 'Babyausstattung' OR
214 $zusatz_leistungsart == 'Babybett' OR $zusatz_leistungsart ==
215 'Highchair' OR $zusatz_leistungsart == 'Hochstuhl' OR
216 $zusatz_leistungsart == 'Kinderbett') {
217             $preis[$i] =
218             $aufenthaltsdauer/7*$_SESSION['person'][12]*$zusatz[$i][2];
219         }
220         else {
221             $preis[$i] = $aufenthaltsdauer/7*$_SESSION['person'][11]*
222             $zusatz[$i][2];
223         }
224     }
225     else {
226         $preis[$i] = $zusatz[$i][2];
227     }
228
229     ?>
230 <br>&nbsp; Gesamt:&nbsp;

```

```

231     <?php
232     if ($zusatz[$i][3] == 'EUR') {
233         echo $preis[$i]. '&nbsp;'. $zusatz[$i][3];
234     }
235     else {
236         $moneur->eurpreis($zusatz[$i][3], $preis[$i]);
237         $europreis = $moneur->preis_eur;
238         echo $preis[$i]. '&nbsp;'. $zusatz[$i][3].
239         '&nbsp;'. $europreis. '&nbsp;EUR';
240     }
241     ?>
242 </td>
243 </tr>
244 <?php
245 $gesamtkosten = $gesamtkosten+$preis[$i];
246 $waehrung = $zusatz[$i][3];
247 $preiseurges = $preiseurges+$europreis;
248 }
249 }
250 ?>
251 <tr>
252 <td>
253 <span class="kursiv">Freiwillige Zusatzleistungen:</span>
254 </td>
255 <td>
256 &nbsp;
257 </td>
258 </tr>
259 <?php
260 $zusatz_berechnung = array ('pro_Tag', 'pro_Woche', 'pro_Person',
261 'pro_Person_Tag', 'pro_Person_Woche');
262 $zusatz = $_SESSION['zusatz'];
263 for ($i=5; $i < 30; $i++) {
264     if (is_array($zusatz[$i])) {
265         $waehrung-alternativ = trim($zusatz[$i][3]);
266         $zusatz_leistungsart = trim($zusatz[$i][1]);
267         $zusatz_berechnung1 = trim($zusatz[$i][5]);
268         $zusatz_berechnung1 = str_replace("/", "_", $zusatz_berechnung1);
269         $zusatz_berechnung1 = str_replace("_", "-", $zusatz_berechnung1);
270     }
271 <tr>
272 <td>
273 <?php echo utf8_decode($zusatz[$i][1]). ',&nbsp;'. $zusatz[$i][2].
274 '&nbsp;'. $zusatz[$i][3]; ?>
275 </td>
276 <td> <?php echo $zusatz[$i][4]. 'x,&nbsp;'. $zusatz[$i][5]; ?>
277 <?php
278 /* Berechnung Pro Woche */
279 if (preg_match("/\b$zusatz_berechnung[1]\b/i", $zusatz_berechnung1)) {
280     $preis[$i] = $aufenthaltsdauer/7*$zusatz[$i][2];
281 }
282 /* Berechnung Pro Tag */
283 elseif (preg_match("/\b$zusatz_berechnung[0]\b/i",
284 $zusatz_berechnung1)) {
285     $preis[$i] = $aufenthaltsdauer*$zusatz[$i][2];
286 }
287 /* Berechnung Pro Person */
288 elseif (preg_match("/\b$zusatz_berechnung[2]\b/i",
289 $zusatz_berechnung1)) {
290     if ($zusatz_leistungsart == 'Babyausstattung' OR
291 $zusatz_leistungsart == 'Babybett' OR $zusatz_leistungsart ==
292 'Highchair' OR $zusatz_leistungsart == 'Hochstuhl' OR
293 $zusatz_leistungsart == 'Kinderbett') {
294         $preis[$i] = $_SESSION['person'][12]*$zusatz[$i][2];
295     }
296     else {

```



```

297         $preis[$i] = $_SESSION['person'][11]*$zusatz[$i][2];
298     }
299 }
300 /* Berechnung Pro Person-Tag */
301 elseif (preg_match("/\b$zusatz_berechnung[3]\b/i",
302 $zusatz_berechnung1)) {
303     if ($zusatz_leistungsart == 'Babyausstattung' OR
304     $zusatz_leistungsart == 'Babybett' OR $zusatz_leistungsart ==
305     'Highchair' OR $zusatz_leistungsart == 'Hochstuhl' OR
306     $zusatz_leistungsart == 'Kinderbett') {
307         $preis[$i] =
308         $aufenthaltsdauer*$_SESSION['person'][12]*$zusatz[$i][2];
309     }
310     else {
311         $preis[$i] = $aufenthaltsdauer*$_SESSION['person'][11]*
312         $zusatz[$i][2];
313     }
314 }
315 /* Berechnung Pro Person-Woche */
316 elseif (preg_match("/\b$zusatz_berechnung[4]\b/i",
317 $zusatz_berechnung1)) {
318     if ($zusatz_leistungsart == 'Babyausstattung' OR
319     $zusatz_leistungsart == 'Babybett' OR $zusatz_leistungsart ==
320     'Highchair' OR $zusatz_leistungsart == 'Hochstuhl' OR
321     $zusatz_leistungsart == 'Kinderbett') {
322         $preis[$i] =
323         $aufenthaltsdauer/7*$_SESSION['person'][12]*$zusatz[$i][2];
324     }
325     else {
326         $preis[$i] = $aufenthaltsdauer/7*$_SESSION['person'][11]*
327         $zusatz[$i][2];
328     }
329 }
330 else {
331     $preis[$i] = $zusatz[$i][2];
332 }
333
334 ?>
335 <br>&nbsp;Gesamt:&nbsp;
336 <?php
337 if ($zusatz[$i][3] == 'EUR') {
338     echo $preis[$i]. '&nbsp;'. $zusatz[$i][3];
339 }
340 else {
341     $moneur->eurpreis($zusatz[$i][3], $preis[$i]);
342     $europreis = $moneur->preis_eur;
343     echo $preis[$i]. '&nbsp;'. $zusatz[$i][3].
344     '&nbsp;/&nbsp;'. $europreis. '&nbsp;EUR';
345 }
346 ?>
347 </td>
348 </tr>
349 <?php
350 $gesamtkosten = $gesamtkosten+$preis[$i];
351 $waehrung = $zusatz[$i][3];
352 $preiseurges = $preiseurges+$europreis;
353
354 }
355 }
356 ?>
357 <tr>
358     <td>
359         <span>Buchungsgeb&#252;hr:</span>
360         <?php $preiseurges = $preiseurges+50;?>
361     </td>
362     <td>

```

```

363         <span>50 EUR</span>
364     </td>
365 </tr>
366 <tr>
367     <td>
368         <span class="fett">Gesamtkosten Zusatzleistungen:</span>
369     </td>
370     <td>
371         <span class="fett">
372             <?php
373                 if ($waehrung_alternativ == 'EUR') {
374                     echo $gesamtkosten.'&nbsp;'.$waehrung;
375                 }
376                 else {
377                     echo $gesamtkosten.'&nbsp;'.$waehrung.
378                         '&nbsp;/'.$waehrung.' $preiseurges.'&nbsp;EUR';
379                 }
380             ?>
381         </span>
382     </td>
383 </tr>
384 <tr>
385     <td>
386         &nbsp;
387     </td>
388     <td>
389         &nbsp;
390     </td>
391 </tr>
392 <tr>
393     <td>
394         <span class="fett">Buchungskosten:</span>
395     </td>
396     <td>
397         <span class="fett"><?php
398             $buchungskosten = $objekt_gesamt+$SESSION['zusatz'][1][0];
399             $buchungskosten = $buchungskosten+50;
400             echo $buchungskosten.'&nbsp;'.$waehrungobj;
401             ?></span>
402     </td>
403 </tr>
404 </table>
405 </div>
406 Die Buchungskosten beziehen sich auf den Objektpreis. Verpflichtende
407 Zusatzleistungen <br>
408 wie z.B. Kautionen.und Zus&#228;tzlich gebuchte Leistungen sind vor Ort zu
409 begleichen.
410
411 <div class="headline">Objekt- und Reisedaten</div>
412 <table>
413     <tr>
414         <td>
415             Gebuchtes Objekt:
416         </td>
417         <td>
418             <?php echo $objektreferenz.'.'. $objektkategorie; ?>
419         </td>
420     </tr>
421     <tr>
422         <td>
423             Anreisedatum:
424         </td>
425         <td>
426             <?php echo $anreisedatum; ?>
427         </td>
428     </tr>

```

```

429     <tr>
430         <td>
431             Aufenthaltsdauer
432         </td>
433         <td>
434             <?php echo $aufenthaltsdauer; ?>&nbsp;Tage
435         </td>
436     </tr>
437 </table>
438 <form action="buchen1.php">
439     <input name="sid" type="hidden" value="<?php echo $_SESSION['_?>">
440     <input type="submit" value="Angaben_korrigieren">
441 </form>
442 <form action="buchen_interhome.php">
443     <input name="sid" type="hidden" value="<?php echo $_SESSION['_?>">
444     <input type="submit" value="Buchen">
445 </form>
446 <?php
447 }
448
449 function zusatzleistungen_wolters ($pfad, $anbieter, $objektreferenz,
450 $objektkategorie, $anreisedatum, $aufenthaltsdauer, $fehler, $sessionid) {
451     $objektpreis = $_SESSION['objekt'][5];
452     $versicherung = $_SESSION['zusatz'][0];
453     $objekt_gesamt = $objektpreis[0] + $versicherung[2];
454     ?>
455     <div class="bu_zubview">
456         <table>
457             <tr>
458                 <td>
459                     Preis des Objektes:
460                 </td>
461                 <td>
462                     <?php echo $objektpreis[0]. '&nbsp;'; $. $objektpreis[1]; ?>
463                 </td>
464             </tr>
465             <tr>
466                 <td>
467                     Versicherung:
468                 </td>
469                 <td>
470                     <?php
471                     if ($versicherung == 'Y') {
472                         echo 'Best&#228;tigt';
473                     }
474                     else {
475                         echo 'Nicht_best&#228;tigt';
476                     }
477                     ?>
478                 </td>
479             </tr>
480             <tr>
481                 <td>
482                     <span class="fett">Objekt Gesamt:</span>
483                 </td>
484                 <td>
485                     <span class="fett"><?php echo $objekt_gesamt. '&nbsp;'; $. $objektpreis[1];
486                     ?></span>
487                 </td>
488             </tr>
489             <tr>
490                 <td>
491                     &nbsp;
492                 </td>
493                 <td>
494                     &nbsp;

```

```

495     </td>
496 </tr>
497 <tr>
498     <td>
499         <span>Verpflichtende Zusatzleistungen:</span>
500     </td>
501     <td>
502         <span>
503             <?php
504                 $gesamtkosten = 0;
505                 if ($_SESSION['zusatz'][1][0] == 0 OR $_SESSION['zusatz'][1][0] == '') {
506                     echo 'keine';
507                 }
508                 else {
509                     echo $_SESSION['zusatz'][1][0]. '&nbsp;';
510                     $gesamtkosten = $gesamtkosten + $_SESSION['zusatz'][1][0];
511                 }
512             ?>
513         </span>
514     </td>
515 </tr>
516 <tr>
517     <td>
518         <span class="kursiv">Freiwillige Zusatzleistungen:</span>
519     </td>
520     <td><span class="kursiv">
521         <?php
522             if (!is_array($_SESSION['zusatz'][5])) {
523                 echo 'keine';
524             }
525             else {
526                 echo 'Gesamt';
527             }
528             ?>
529         </span>
530     </td>
531 </tr>
532 <?php
533 $zusatz_berechnung = array ('Aufenthalt', 'Woche', 'Tag', 'AufenthaltxEinheit',
534 'WochexEinheit', 'TagxEinheit');
535 $zusatz = $_SESSION['zusatz'];
536 for ($i=5; $i < 30; $i++) {
537     if (is_array($zusatz[$i])) {
538         $zusatz_leistungsart = trim($zusatz[$i][1]);
539         $zusatz_berechnung1 = trim($zusatz[$i][5]);
540         $zusatz_berechnung1 = str_replace("/", "x", $zusatz_berechnung1);
541         ?>
542     <tr>
543     <td>
544         <?php
545             echo $zusatz[$i][1]. '&nbsp;';
546             echo $zusatz[$i][3]. '&nbsp;';
547             echo $zusatz[$i][4]. 'x';
548             echo $zusatz[$i][5];
549         </td>
550     <td>
551         <?php
552             /* Berechnung Pro Woche */
553             if (preg_match("/\b$zusatz_berechnung[1]\b/i", $zusatz_berechnung1)) {
554                 $preis[$i] = $aufenthaltsdauer/7*$zusatz[$i][2];
555             }
556             <?php
557                 echo $preis[$i]. '&nbsp;';
558                 echo $zusatz[$i][3];
559             </td>
560 </tr>
561 <?php

```

```

561     $gesamtkosten = $gesamtkosten+$preis[$i];
562     $waehrung = $zusatz[$i][3];
563 }
564 /* Berechnung Pro Tag */
565 elseif (preg_match("/\b$zusatz_berechnung[2]\b/i", $zusatz_berechnung1)) {
566     $preis[$i] = $aufenthaltsdauer*$zusatz[$i][2];
567     ?>
568     <?php echo $preis[$i]. '&nbsp;'. $zusatz[$i][3]; ?>
569     </td>
570 </tr>
571 <?php
572     $gesamtkosten = $gesamtkosten+$preis[$i];
573     $waehrung = $zusatz[$i][3];
574 }
575 /* Berechnung Pro Aufenthalt/Einheit */
576 elseif (preg_match("/\b$zusatz_berechnung[3]\b/i", $zusatz_berechnung1)) {
577     $preis[$i] = $zusatz[$i][2]*$zusatz[$i][4];
578     ?>
579     <?php echo $preis[$i]. '&nbsp;'. $zusatz[$i][3]; ?>
580     </td>
581 </tr>
582 <?php
583     $gesamtkosten = $gesamtkosten+$preis[$i];
584     $waehrung = $zusatz[$i][3];
585 }
586 /* Berechnung Pro Woche/Einheit */
587 elseif (preg_match("/\b$zusatz_berechnung[4]\b/i", $zusatz_berechnung1)) {
588     $preis[$i] = $aufenthaltsdauer/7*$zusatz[$i][2]*$zusatz[$i][4];
589     ?>
590     <?php echo $preis[$i]. '&nbsp;'. $zusatz[$i][3]; ?>
591     </td>
592 </tr>
593 <?php
594     $gesamtkosten = $gesamtkosten+$preis[$i];
595     $waehrung = $zusatz[$i][3];
596 }
597 /* Berechnung Pro Tag/Einheit */
598 elseif (preg_match("/\b$zusatz_berechnung[5]\b/i", $zusatz_berechnung1)) {
599     $preis[$i] = $aufenthaltsdauer*$zusatz[$i][2]*$zusatz[$i][4];
600     ?>
601     <?php echo $preis[$i]. '&nbsp;'. $zusatz[$i][3]; ?>
602     </td>
603 </tr>
604 <?php
605     $gesamtkosten = $gesamtkosten+$preis[$i];
606     $waehrung = $zusatz[$i][3];
607 }
608 else {
609     ?>
610     </td>
611 </tr>
612 <?php
613     $gesamtkosten = $gesamtkosten+$zusatz[$i][2];
614     $waehrung = $zusatz[$i][3];
615 }
616 }
617 }
618 ?>
619 <tr>
620 <td>
621     <span class="fett">Gesamtkosten Zusatzleistungen:</span>
622 </td>
623 <td><span class="fett">
624 <?php
625     if ($gesamtkosten == '' OR $gesamtkosten == 0) {
626         echo 'keine';

```

```

627     }
628     else {
629         echo $gesamtkosten.'&nbsp;'.$waehrung;
630     }
631     ?>
632     </span>
633 </td>
634 </tr>
635 <tr>
636     <td>
637         &nbsp;
638     </td>
639     <td>
640         &nbsp;
641     </td>
642 </tr>
643 <tr>
644     <td>
645         <span class="fett">Gesamtkosten Buchung:</span>
646     </td>
647     <td>
648         <span class="fett">
649             <?php
650                 $allekosten = $gesamtkosten+$objekt_gesamt;
651                 echo $allekosten.'&nbsp;'.$waehrung;
652             ?>
653         </span>
654     </td>
655 </tr>
656 </table>
657 </div>
658 <span>Achtung, die von Ferienhaus.de errechneten Werte sind nur
659 N&#228;herungswerte.<br>Es werden u.a. die Kosten f&#252;r die Reise
660 R&#252;cktritts Versicherung nicht berechnet, da sie<br>Ferienhaus.de nicht
661 vorliegen</span>
662 <div class="headline">Objekt und Reisedaten</div>
663 <table>
664     <tr>
665         <td>
666             Objekt:
667         </td>
668         <td>
669             <?php echo $objektreferenz.' '.$objektkategorie; ?>
670         </td>
671     </tr>
672     <tr>
673         <td>
674             Anreisedatum:
675         </td>
676         <td>
677             <?php echo $anreisedatum; ?>
678         </td>
679     </tr>
680     <tr>
681         <td>
682             Aufenthaltsdauer:
683         </td>
684         <td>
685             <?php echo $aufenthaltsdauer.'&nbsp;Tage'; ?>
686         </td>
687     </tr>
688 </table>
689 <div class="headline">Best&#228;tigung</div>
690 <form action="buchen1.php">
691     <input name="sid" type="hidden" value="<?php echo $_SESSIONID; ?>">
692     <input type="submit" value="Angaben korrigieren">

```

```

693 </form>
694 <form action="buchen_wolters.php">
695     <input name="sid" type="hidden" value="<?php_echo_.$sessionid;_?>">
696     <input type="submit" value="Buchen">
697 </form>
698 <?php
699 }
700 ?>
701
702 <html>
703 <html xmlns="http://www.w3.org/1999/xhtml">
704 <head>
705     <meta http-equiv="Content-Type" content="text/html;_charset=UTF-8">
706     <title>Abfragetests</title>
707     <link href="../../css/ferienhaus_ibe.css" rel="stylesheet" type="text/css">
708 </head>
709 <body>
710 <!-- ***Hauptcontainer Start*** //—>
711 <div id="main">
712     <div class="bu_header"> <span>Adressdaten</span>&nbsp; ||&nbsp;  
713     <span>Zahlungsangaben</span>&nbsp; ||&nbsp;  <span class="red">Angaben
714     pruefen</span>&nbsp; ||&nbsp;  Buchung best&#228;tigen</div>
715 <?php
716     require_once '../header.php';
717     ?>
718     <div class="headline">Ihre Daten</div>
719     <div class="bu_persodat1">
720         <table>
721             <tr>
722                 <td>
723                     <span class="fett">Anrede:</span>
724                 </td>
725                 <td>
726                     <?php
727                         if ($.SESSION['person'][0] == 1) {
728                             echo 'Herr';
729                         }
730                         else {
731                             echo 'Frau';
732                         }
733                     ?>
734                 </td>
735             <tr>
736                 <tr>
737                     <td>
738                         <span class="fett">Name:</span>
739                     </td>
740                     <td>
741                         <?php echo $.SESSION['person'][1]; ?>
742                     </td>
743                 <tr>
744                 <tr>
745                     <td>
746                         <span class="fett">Vorname</span>
747                     </td>
748                     <td>
749                         <?php echo $.SESSION['person'][2]; ?>
750                     </td>
751                 <tr>
752                 <tr>
753                     <td>
754                         <span class="fett">Strasse:</span>
755                     </td>
756                     <td>
757                         <?php echo $.SESSION['person'][3]; ?>
758                     </td>

```

```

759     <tr>
760     <tr>
761         <td>
762             <span class="fett">Postleitzahl:</span>
763         </td>
764     <td>
765         <?php echo $_SESSION['person'][4]; ?>
766     </td>
767 </tr>
768 <tr>
769     <td>
770         <span class="fett">Ort:</span>
771     </td>
772     <td>
773         <?php echo $_SESSION['person'][5]; ?>
774     </td>
775 </tr>
776 <tr>
777     <td>
778         <span class="fett">Herkunftsland:</span>
779     </td>
780     <td>
781         <?php
782         $idland = $_SESSION['person'][6];
783         if ($idland != '') {
784             $query = "SELECT_Land_FROM_ref_land_WHERE_IDLand=_ '$idland' _LIMIT_1";
785             $result = mysql_query($query);
786             while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
787                 $land = $line['Land'];
788             }
789             echo $land;
790         }
791         ?>
792     </td>
793 </tr>
794 <tr>
795     <td>
796         <span class="fett">Telefon:</span>
797     </td>
798     <td>
799         <?php echo $_SESSION['person'][7]; ?>
800     </td>
801 </tr>
802 </table>
803 </div>
804 <div class="bu_persodat2">
805     <table>
806     <tr>
807     <td>
808         <span class="fett">Fax, Tel2:</span>
809     </td>
810     <td>
811         <?php echo $_SESSION['person'][8]; ?>
812     </td>
813 </tr>
814 <tr>
815     <td>
816         <span class="fett">Handy:</span>
817     </td>
818     <td>
819         <?php echo $_SESSION['person'][9]; ?>
820     </td>
821 </tr>
822 <tr>
823     <td>
824         <span class="fett">E-Mail:</span>

```



```

825         </td>
826     <td>
827         <?php echo $_SESSION[ 'person' ][10]; ?>
828     </td>
829 </tr>
830 <tr>
831     <td>
832         <span class="fett">Anz. Reisender:</span>
833     </td>
834     <td>
835         <?php echo $_SESSION[ 'person' ][11]; ?>
836     </td>
837 </tr>
838 <tr>
839     <td>
840         <span class="fett">Davon Kinder:</span>
841     </td>
842     <td>
843         <?php
844             if ( $_SESSION[ 'person' ][12] == '' OR $_SESSION[ 'person' ][12] == '0' ) {
845                 echo 'Keine';
846             }
847             else {
848                 echo $_SESSION[ 'person' ][12];
849             }
850             ?>
851     </td>
852 </tr>
853 <?php
854     /* Vorerst nur für Wolters wichtig */
855     if ( $anbieter == 2 ) {
856         ?>
857     <tr>
858         <td>
859             <span class="fett">Bemerkung:</span>
860         </td>
861         <td>
862             <?php
863                 $bemerkung = $_SESSION[ 'person' ][13];
864                 if ( strlen( $bemerkung ) > 100 ) {
865                     $bemerkung = substr( $bemerkung, 0, 100 ). '...';
866                     echo $bemerkung;
867                 }
868                 else {
869                     echo $bemerkung;
870                 }
871                 ?>
872         </td>
873     </tr>
874     <tr>
875         <td>
876             <span class="fett">Buchungshinweise:</span>
877         </td>
878         <td>
879             <?php
880                 $buhinweise = $_SESSION[ 'person' ][14];
881                 if ( $buhinweise == 'Y' ) {
882                     echo 'Best&#228;tigt';
883                 }
884                 else {
885                     echo 'Nicht&#228;tigt';
886                 }
887                 ?>
888         </td>
889     </tr>
890 <?php

```

```

891     }
892     ?>
893     <tr>
894         <td>
895             <span class="fett">AGBs:</span>
896         </td>
897     </tr>
898     <?php
899         if ($_SESSION['person'][15] == 'Y') {
900             echo 'Best&#228;tigt';
901         }
902         else {
903             echo 'nicht_Best&#228;tigt';
904             $halt = 1;
905         }
906     ?>
907 </td>
908 </tr>
909 </table>
910 </div>
911 <div class="headline">Zusatzleistungen und Zahlungsdetails</div>
912 <div class="bu_payview">
913 <?php
914 if ($_SESSION['payment'][0] == 1) {
915     $bez = $_SESSION['payment'][1];
916     ?>
917     <h3>Zahlung per Bankeinzug</h3>
918     <table>
919         <tr>
920             <td>
921                 Kontonummer:
922             </td>
923             <td>
924                 <?php echo $bez[0]; ?>
925             </td>
926         </tr>
927         <tr>
928             <td>
929                 Bankleitzahl:
930             </td>
931             <td>
932                 <?php echo $bez[1]; ?>
933             </td>
934         </tr>
935         <tr>
936             <td>
937                 Bank:
938             </td>
939             <td>
940                 <?php echo $bez[2]; ?>
941             </td>
942         </tr>
943     </table>
944     <?php
945 }
946 elseif ($_SESSION['payment'][0] == 2) {
947     $cc = $_SESSION['payment'][2];
948     ?>
949     <h3>Zahlung mit Kreditkarte</h3>
950     <table>
951         <tr>
952             <td>
953                 Karte:
954             </td>
955             <td>
956                 <?php

```

```

957     $query = "SELECT typ FROM mon_creditcard WHERE kurz = '$cc[0]' LIMIT 1";
958     $result = mysql_query($query);
959     while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
960         $skarte = $line['typ'];
961     }
962     echo $skarte;
963     ?>
964 </td>
965 </tr>
966 <tr>
967     <td>
968         Kartennummer:
969     </td>
970     <td>
971         <?php echo $cc[1]; ?>
972     </td>
973 </tr>
974 <tr>
975     <td>
976         Verifizierung:
977     </td>
978     <td>
979         <?php echo $cc[2]; ?>
980     </td>
981 </tr>
982 <tr>
983     <td>
984         CC#252;ltigkeit:
985     </td>
986     <td>
987         <?php echo $cc[4]. '/' . $cc[3]; ?>
988     </td>
989 </tr>
990 <tr>
991     <td>
992         Karteninhaber:
993     </td>
994     <td>
995         <?php echo $cc[6]. '&nbsp;' . $cc[5]; ?>
996     </td>
997 </tr>
998 </table>
999 <?php
1000 }
1001 elseif ($.SESSION['payment'][0] == 3) {
1002     ?>
1003     <h3>Zahlung auf Rechnung</h3>
1004     <table>
1005     <tr>
1006     <td>
1007         Rechnung:
1008     </td>
1009     <td>
1010         Zahlung auf Rechnung
1011     </td>
1012 </tr>
1013 </table>
1014 <?php
1015 }
1016 ?>
1017 </div>
1018 <?php
1019 if ($anbieter == '1') {
1020     call_user_func('zusatzleistungen_interhome',$pfad, $anbieter, $objektreferenz,
1021     $objektkategorie, $anreisedatum, $aufenthaltsdauer, $fehler, $sessionid);
1022 }

```

```

1023     elseif ($anbieter == '2') {
1024         call_user_func('zusatzleistungen_wolters',$pfad, $anbieter, $objektreferenz,
1025             $objektkategorie, $anreisedatum, $aufenthaltsdauer, $fehler, $sessionid);
1026     }
1027     ?>
1028 </div>
1029 <!-- ***Hauptcontainer Ende*** //—>
1030 </body>
1031 </html>

```

G.14. Datenversand Interhome

Listing G.12: Quelltext Datenversand Interhome

```

1 <?php
2
3 /**
4  * Übermittlung der Daten an die Interhome
5  * Buchungsschnittstelle mit vorheriger
6  * Aufbereitung der Daten
7  */
8
9 $sessionid = $_REQUEST['sid'];
10 if (!isset($_COOKIE['PHPSESSID'])) {
11     session_id($sessionid);
12 }
13 session_start();
14 $tabelle = 'buc_personen';
15 $datum_aktuell = date("Y-m-d");
16 $zeit_aktuell = date("H:i:s");
17
18 /* Positionen: 36*/
19
20 /* Abfrageschnittstellen einbinden */
21 $pfad = getenv("DOCUMENTROOT");
22 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
23 include $datenbankzugriff;
24 $inc_objektklasse = $pfad.'/include/class_objekte_db.php';
25 include $inc_objektklasse;
26 $inc_datumsklasse = $pfad.'/include/class_datumfkt.php';
27 include $inc_datumsklasse;
28
29 /* Interhome SOAP Header laden */
30 $interhome_soap_connect = $pfad.'/include/interhome_soapconnect.php';
31 include $interhome_soap_connect;
32
33 /* Abfrageschnittstellen initialisieren */
34 $objektdaten = new datensaetze;
35 $datumfkt = new zeitfunktionen;
36
37 /* Objektdaten referenzieren */
38 $objektreferenz = $_SESSION['objekt'][0];
39 $objektkategorie = $_SESSION['objekt'][1];
40 $anreisedatum = $_SESSION['objekt'][2];
41 $aufenthaltsdauer = $_SESSION['objekt'][3];
42 $anbieter = $_SESSION['objekt'][4];
43
44 $datumfkt->datum_deisoun($anreisedatum);
45 $startdatum_iso = $datumfkt->datum_iso;
46 $startdatum_unix = $datumfkt->datum_unix;
47
48 $duration = $aufenthaltsdauer*86400;
49 $enddatum_unix = $startdatum_unix+$duration;
50 $enddatum_iso = strftime("%Y-%m-%d", $enddatum_unix);

```

```

51
52 $pets = 'N';
53 for ($i = 5; $i < 30; $i++) {
54     if (is_array($_SESSION['zusatz'][$i])) {
55         $position = $_SESSION['zusatz'][$i][0];
56         $beschreibung = $_SESSION['zusatz'][$i][1];
57         $anzahl = $_SESSION['zusatz'][$i][4];
58         if ($beschreibung == 'Haustier') { //Haustier reist mit
59             $pets = 'Y';
60         }
61     }
62 }
63
64 $extrabed = '';
65 $extrabed1 = '';
66 $extrabed2 = '';
67 for ($i = 5; $i < 30; $i++) {
68     if (is_array($_SESSION['zusatz'][$i]) AND $_SESSION['zusatz'][$i][1] != '') {
69         if ($_SESSION['zusatz'][$i][0] == 'Haustier') {
70             $pets = 'Y';
71         }
72
73         if ($_SESSION['zusatz'][$i][0] == 'Zusatzbett') {
74             $extrabed1 = 'A';
75             $extrabed = 'A';
76         }
77
78         if ($_SESSION['zusatz'][$i][0] == 'Kinderbett') {
79             $extrabed2 = 'C';
80             $extrabed = 'C';
81         }
82
83         if ($extrabed1 == 'A' AND $extrabed2 == 'C') {
84             $extrabed = 'B';
85         }
86     }
87 }
88
89 $so = '5059';
90 $nref = $_SESSION['objekt'][0].'.'. $_SESSION['objekt'][1]; /* Referenznummer des
91 Objektes, z.B. "AT6280.300.1" */
92 $checkin = $startdatum_iso; /* Anreisedatum */
93 $checkout = $enddatum_iso; /* Abreisedatum */
94 $languagecode = 'DE'; /* Sprachschlüssel */
95 // $extrabed = ''; /* Extra Bett, Kinderbett oder Bett */
96 // $pets = ''; /* Haustier reist mit */
97 $occupantsadt = ''; /* Nicht benötigt */
98 $occupantschd = ''; /* Nicht benötigt */
99 $occupantsinf = ''; /* Nicht benötigt */
100 $insains = '';
101 if ($_SESSION['zusatz'][0][0] != 'KEIN' AND $_SESSION['zusatz'][0][0] != '') {
102     $insains = 'Y'; /* Reise Rücktrittsversicherung */
103 }
104 $insextracode = $_SESSION['zusatz'][0][0]; /* Art der Reiserücktrittsversicherung */
105 $inschilds = ''; /* Nur in Sales Office (SO) 30 & 31, DE = 59 */
106 $clientfoa = $_SESSION['person'][0]; /* Anrede */
107 $clientfname = $_SESSION['person'][2]; /* Vorname */
108 $clientlname = $_SESSION['person'][1]; /* Nachname */
109 $clientadr1 = $_SESSION['person'][3]; /* Strasse */
110 $clientadr2 = ''; /* Adresszusatz */
111 $clientzip = $_SESSION['person'][4]; /* Postleitzahl */
112 $clientplace = $_SESSION['person'][5]; /* Wohnort */
113 $clientstate = ''; /* Bundesland */
114 $clientcountry = $_SESSION['person'][6]; /* Land, 2 Buchstabencode */
115 $clienttel = $_SESSION['person'][7]; /* Telefon */
116 $clientfax = $_SESSION['person'][8]; /* Faxnummer */

```

```

117 $clientemail = $_SESSION['person'][10]; /* E-Mail Adresse */
118 $siteid = ''; /* Nicht benötigt */
119 $partnerid = 'DE591104'; /* ID des Anbieters der vermittelt, hier Online Now mit
120 XXXXXXXX */
121 $partneridzu = ''; /* Zusatz ID für Events und Specials seitens des Vermittlers */
122 $source = ''; /* Nicht benötigt */
123
124 /* Zahlungsart */
125 $paymode = $_SESSION['payment'][0];
126 if ($paymode == 2) {
127     $paymode_ih = 1;
128 }
129 elseif ($paymode == 3) {
130     $paymode_ih = 4;
131 }
132 $creditcardtype = $_SESSION['payment'][2][0]; /* Kreditkartentyp */
133 $creditcardnumber = $_SESSION['payment'][2][1]; /* Kreditkartennummer */
134 $creditcardcvc = $_SESSION['payment'][2][2]; /* Kreditkarten Verifikationscode */
135 $creditcardexpiry = $_SESSION['payment'][2][3].$_SESSION['payment'][2][4];
136 /* Ablaufdatum Kreditkarte */
137 $creditcardholder = $_SESSION['payment'][2][5]. '&nbsp;'. $_SESSION['payment'][2][6];
138 /* Kreditkartenbesitzer */
139 $environment = ''; /* Abgeschaltet, ehemals Schalter für Festlegung der Ausführenden
140 Umgebung seitens Interhome */
141
142 if ($paymode_ih == 1) {
143     $params = array('So' => $so, 'Nref' => $nref, 'CheckIn' => $checkin, 'CheckOut' =>
144     $checkout,
145     'LanguageCode' => $languagecode, 'ExtraBed' => $extrabed, 'Pets' => $pets,
146     'OccupantsAdt' => $occupantsad, 'OccupantsChd' => $occupantschd, 'OccupantsInf'
147     => $occupantsinf, 'InsAins' => $insains, 'InsExtraCode' => $insextracode,
148     'InsChilds' => $inschilds, 'ClientFoa' => $clientfoa, 'ClientFname' =>
149     $clientfname, 'ClientLname' => $clientlname,
150     'ClientAdr1' => $clientadr1, 'ClientAdr2' => $clientadr2, 'ClientZip' =>
151     $clientzip, 'ClientPlace' => $clientplace,
152     'ClientState' => $clientstate, 'ClientCountry' => $clientcountry, 'ClientTel' =>
153     $clienttel, 'ClientFax' => $clientfax,
154     'ClientEmail' => $clientemail, 'SiteID' => $siteid, 'PartnerID' => $partnerid,
155     'PartnerIDZU' => $partneridzu,
156     'Source' => $source, 'PayMode' => $paymode_ih, 'CreditCardType' =>
157     $creditcardtype, 'CreditCardNumber' => $creditcardnumber,
158     'CreditCardCvc' => $creditcardcvc, 'CreditCardExpiry' => $creditcardexpiry,
159     'CreditCardHolder' => $creditcardholder, 'Environment' => $environment);
160 }
161 elseif ($paymode_ih == 4) {
162     $params = array('So' => $so, 'Nref' => $nref, 'CheckIn' => $checkin, 'CheckOut' =>
163     $checkout,
164     'LanguageCode' => $languagecode, 'ExtraBed' => $extrabed, 'Pets' => $pets,
165     'OccupantsAdt' => $occupantsad, 'OccupantsChd' => $occupantschd, 'OccupantsInf'
166     => $occupantsinf, 'InsAins' => $insains, 'InsExtraCode' => $insextracode,
167     'InsChilds' => $inschilds, 'ClientFoa' => $clientfoa, 'ClientFname' =>
168     $clientfname, 'ClientLname' => $clientlname,
169     'ClientAdr1' => $clientadr1, 'ClientAdr2' => $clientadr2, 'ClientZip' =>
170     $clientzip, 'ClientPlace' => $clientplace,
171     'ClientState' => $clientstate, 'ClientCountry' => $clientcountry, 'ClientTel' =>
172     $clienttel, 'ClientFax' => $clientfax,
173     'ClientEmail' => $clientemail, 'SiteID' => $siteid, 'PartnerID' => $partnerid,
174     'PartnerIDZU' => $partneridzu,
175     'Source' => $source, 'PayMode' => $paymode_ih, 'CreditCardType' => '',
176     'CreditCardNumber' => '',
177     'CreditCardCvc' => '', 'CreditCardExpiry' => '', 'CreditCardHolder' => '',
178     'Environment' => $environment);
179 }
180
181 $result = $client->__soapCall("PriceRqst", array('parameters'
182 => $params), null, $header);

```

```

183 print_r($params);
184 echo '<br>';
185 print_r($result);
186 echo '<br><br>';
187 $waehrung = $result->PriceRqstResult->Currency;
188 $objektpreis = $result->PriceRqstResult->PriceRent;
189 $buchgebuehr = $result->PriceRqstResult->PriceBookingFee;
190 $anzahlung = $result->PriceRqstResult->PricePrePayment;
191 $sende_anzahlung = $result->PriceRqstResult->ExpirationDatePrePayment;
192 $sende_restzahlung = $result->PriceRqstResult->ExpirationDateResidue;
193 $gesamtzahlung = $result->PriceRqstResult->PriceClientTotal;
194
195 /* Unterscheidung einfügen, buchungserfolg =Y wenn Erfolg, sonst buchungserfolg =
196 N—> */
197 $query = "INSERT INTO _$tabelle
198 (id ,name,vorname, str ,plz ,ort ,tel ,email ,anbieter ,objekt ,kategorie ,datum ,zeit ,buchungse
199 rfolg ) VALUES (',' , '$clientlname ' , _ '$clientfname ' , _ '$clientadr1 ' , '$clientzip ' ,
200 ' $clientplace ' , _ '$clienttel ' , _ '$clientemail ' , _ '$anbieter ' , ' $objektreferenz ' ,
201 ' $objektkategorie ' , '$datum_aktuell ' , _ '$zeit_aktuell ' , 'Y')";
202 $result = mysql_query($query);
203 if ($result) {
204     echo 'Datensatz _eingefügt ' ;
205 }
206 else {echo mysql_error(); }
207 ?>
208 <html>
209     <html xmlns=" http://www.w3.org/1999/xhtml">
210     <head>
211         <meta http-equiv=" Content-Type" content=" text/html; _charset=UTF-8">
212         <title >Abfragetests</title>
213         <link href=" ../ ../ css/ferienhaus_ibe.css" rel="stylesheet" type=" text/css">
214     </head>
215     <body>
216         <!-- ***Hauptcontainer Start*** //—>
217         <div id="main">
218             <div class="bu_header">
219                 <span>Adressdaten</span>&nbsp;||&nbsp;<span>Zahlungsangaben</span>
220                 &nbsp;||&nbsp;<span>Angaben pruefen</span>&nbsp;||&nbsp;<span>
221                 class="red">Status</span></div>
222                 <?php
223                 require_once ' ../ header.php ' ;
224                 ?>
225                 <div class="headline">Buchungsstatus</div>
226                 Die Buchung Wurde &#252;bermittelt. Der Objektpreis betr&#228;gt <?php echo
227                 $objektpreis.'&nbsp;';.$waehrung; ?><br>
228                 Die Anzahlung von <?php echo $anzahlung.'&nbsp;';.$waehrung; ?> ist bis zum <?php
229                 echo $sende_anzahlung; ?> zu begleichen<br>
230                 Der Restbetrag muss bis zum <?php echo $sende_restzahlung; ?> gezahlt sein.<br>
231                 Durch die Buchungsgeb&#252;hr von <?php echo $buchgebuehr.'&nbsp;';.$waehrung; ?>
232                 betr&#228;gt der Gesamtpreis <?php echo $gesamtzahlung.'&nbsp;';.$waehrung; ?>.
233                 <br>
234                 <a href=" ../ index.php">Zur&#252;ck zur Suchmaske</a><a
235                 href=" buchen1.php">Zur&#252;ck zum Buchungsformular</a>
236             </div>
237             <!-- ***Hauptcontainer Ende*** //—>
238         </body>
239     </html>

```

G.15. Datenversand Wolters

Listing G.13: Quelltext Datenversand Wolters

```

1 <?php
2 /**

```

```

3  * Übermittlung der Daten an die Wolters Buchungsschnittstelle
4  * mit vorheriger Bearbeitung der Daten und Konvertierung in ein
5  * URL konformes Format
6  */
7
8  /* 35 Positionen, 3 freiwillige */
9  /* bei Bankeinzug, 30, 3 freiw. */
10 /* bei Kreditk, 32, 3 freiw */
11
12 $sessionid = $_REQUEST['sid'];
13 if (!isset($_COOKIE['PHPSESSID'])) {
14     session_id($sessionid);
15 }
16 session_start();
17
18 $datum_aktuell = date("Y-m-d");
19 $zeit_aktuell = date("H:i:s");
20 $tabelle = buc_personen;
21
22 /* Abfrageschnittstellen einbinden */
23 $pfad = getenv("DOCUMENTROOT");
24 $datenbankzugriff = $pfad.'/include/opendb_ferienhaus.php';
25 include $datenbankzugriff;
26 $inc_objektklasse = $pfad.'/include/class_objekte_db.php';
27 include $inc_objektklasse;
28 $inc_geldklasse = $pfad.'/include/class_geld.php';
29 include $inc_geldklasse;
30
31 /* Abfrageschnittstellen initialisieren */
32 $objektdaten = new datensaetze;
33 $moneur = new changeto_eur;
34
35 /* Objektdaten referenzieren */
36 $objektreferenz = $_SESSION['objekt'][0];
37 $objektkategorie = $_SESSION['objekt'][1];
38 $anreisedatum = $_SESSION['objekt'][2];
39 $aufenthaltsdauer = $_SESSION['objekt'][3];
40 $anbieter = $_SESSION['objekt'][4];
41 $statusmeldung = 0;
42 /* Personendaten */
43 if ($_SESSION['person'][0] == 1) {
44     $anrede = 'H';
45 }
46 else {
47     $anrede = 'F';
48 }
49
50 $vorname = urlencode($_SESSION['person'][2]);
51 $name = urlencode($_SESSION['person'][1]);
52 $strasse = urlencode($_SESSION['person'][3]);
53 $plz = urlencode($_SESSION['person'][4]);
54 $ort = urlencode($_SESSION['person'][5]);
55 $land_array = array('AT', 'AU', 'BE', 'BG', 'CA', 'CH', 'CS', 'CY', 'CZ', 'DE',
56 'DK', 'ES', 'FIN', 'FR', 'GB', 'GR', 'HR', 'HU', 'IE', 'IT', 'LT', 'MT', 'NL',
57 'NOR', 'PL', 'PT', 'RU', 'SE', 'SI', 'SK', 'TR', 'US');
58 $landwo_array = array('A', 'AU', 'B', 'BG', 'CA', 'CH', 'CS', 'CY', 'CZ', 'D', 'DK',
59 'E', 'FIN', 'F', 'GB', 'GR', 'HR', 'H', 'IRL', 'I', 'LT', 'MT', 'NL', 'N', 'PL',
60 'P', 'RU', 'S', 'SI', 'SK', 'TR', 'US');
61
62 $i = 0;
63 foreach ($land_array AS $land_kz) {
64     if ($_SESSION['person'][6] == $land_kz ) {
65         $land = $landwo_array[$i];
66         if ($landwo_array[$i] == 'D' OR $landwo_array[$i] == 'NL' OR $landwo_array[$i]
67             == 'F') {
68             $sprache = $landwo_array[$i];

```



```

69     }
70     else {
71         $sprache = 'D';
72     }
73 }
74 $i++;
75 }
76
77 $tel1 = urlencode($_SESSION['person'][7]);
78 $tel2 = urlencode($_SESSION['person'][8]);
79 $tel3 = urlencode($_SESSION['person'][9]);
80 $mail = urlencode($_SESSION['person'][10]);
81 $personen = urlencode($_SESSION['person'][11]);
82 $kinder = urlencode($_SESSION['person'][12]);
83 $bemerkung = urlencode($_SESSION['person'][13]);
84 if ($_SESSION['person'][14] == 'Y' AND $_SESSION['person'][15] == 'Y') {
85     $buhinweis = 'yes';
86     $agb = 'yes';
87 }
88 else {
89     $buhinweis = 'no';
90     $agb = 'no';
91 }
92
93 /* Objektdaten */
94 $referenz = $_SESSION['objekt'][0];
95 $kategorie = $_SESSION['objekt'][1];
96 $anreise = $_SESSION['objekt'][2];
97 $aufenthalt = $_SESSION['objekt'][3]/7;
98
99
100 /* Zusatzleistungen */
101 if ($_SESSION['zusatz'][0] == 'Y') {
102     $rrkv = 'yes';
103 }
104 else {
105     $rrkv = 'no';
106 }
107
108 if (is_array($_SESSION['zusatz'][4])) {
109
110     if ($_SESSION['zusatz'][4][4] == 1) {
111         $pet = 'yes';
112     }
113     elseif ($_SESSION['zusatz'][4][4] == 2) {
114         $pet = '2';
115     }
116     elseif ($_SESSION['zusatz'][4][4] == 3) {
117         $pet = '3';
118     }
119     else {
120         $pet = 'no';
121     }
122 }
123 else {
124     $pet = 'no';
125 }
126
127 $urlzusatz = '';
128 for ($i = 5; $i < 30; $i++) {
129     if (is_array($_SESSION['zusatz'][$i])) {
130         $position = $_SESSION['zusatz'][$i][0];
131         $anzahl = $_SESSION['zusatz'][$i][4];
132         $query = "SELECT c_wolters FROM ref_zusatzleistungen WHERE IDZusatz = $position
133         LIMIT 1";
134         $result = mysql_query($query);

```

```

135     while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
136         $c_wolters = $line['c_wolters'];
137     }
138     $urlzusatz = $urlzusatz.'&zub[]='.$referenz.'_'.$c_wolters.'_'.$anzahl;
139 }
140 }
141
142
143 /* Buchen mit Bezahlungsart "Bankeinzug" */
144 if ($_SESSION['payment'][0] == 1) {
145     $kto = urlencode($_SESSION['payment'][1][0]);
146     $blz = urlencode($_SESSION['payment'][1][1]);
147     $bank = urlencode($_SESSION['payment'][1][2]);
148     $zahlart = 'Bankcollection';
149
150     $url = 'http://tui.wol.de/xml_booking.php?objekt_nr='.$
151     $referenz.'&category_id='.$kategorie.
152     '&dummya=53088&datum_kz='.$anreise.
153     '&dauer_kz='.$aufenthalt.'&pers_kz='.$personen.
154     '&kind_kz='.$kinder.'&dummys='.$sprache.'&anrede='.$anrede.'&vorname='.$vorname.
155     '&nachname='.$name.'&strasse='.$strasse.'&plz='.$plz.'&ort='.$ort.'&land='.$land.
156     '&email='.$mail.'&tel_office='.$stell.'&tel_private='.$stel2.'&tel_mobile='.$stel3.
157     '&mitteilung='.$bemerkung.'&payment='.$zahlart.'&kto='.$kto.'&blz='.$
158     $blz.'&bank='.$bank.'&insurance='.$rrkv.'&pet='.$pet.'&travel_conditions='.$
159     $agb.$urlzusatz;
160
161     echo $url;
162 }
163
164 if ($_SESSION['payment'][0] == 2) {
165     $kreditkarte = urlencode($_SESSION['payment'][2][0]);
166     $ccexpirey = urlencode($_SESSION['payment'][2][3]);
167     $ccexpirem = urlencode($_SESSION['payment'][2][4]);
168     $ccnum = urlencode($_SESSION['payment'][2][1]);
169     $ccfname = urlencode($_SESSION['payment'][2][5]);
170     $cclname = urlencode($_SESSION['payment'][2][6]);
171     $zahlart = 'CreditCard';
172
173     $url = 'http://tui.wol.de/xml_booking.php?objekt_nr='.$referenz.
174     '&category_id='.$kategorie.
175     '&dummya=53088&datum_kz='.$anreise.
176     '&dauer_kz='.$aufenthalt.'&pers_kz='.$personen.
177     '&kind_kz='.$kinder.'&dummys='.$sprache.'&anrede='.$anrede.'&vorname='.$vorname.
178     '&nachname='.$name.'&strasse='.$strasse.'&plz='.$plz.'&ort='.$ort.'&land='.$land.
179     '&email='.$mail.'&tel_office='.$stell.'&tel_private='.$stel2.'&tel_mobile='.$
180     $stel3.'&mitteilung='.$bemerkung.'&payment='.$zahlart.'&cc_name='.$
181     $kreditkarte.'&cc_date='.$ccexpirey.
182     '-'.$ccexpirem.'&cc_nr='.$ccnum.'&cc_fn='.$
183     $ccfname.'&cc_ln='.$cclname.'&insurance='.$rrkv.
184     '&pet='.$pet.'&travel_conditions='.$agb.$urlzusatz;
185     echo $url;
186 }
187
188 if ($_SESSION['payment'][0] == 3) {
189     $zahlart = 'Invoice';
190
191     $url = 'http://tui.wol.de/xml_booking.php?objekt_nr='.$referenz.
192     '&category_id='.$kategorie.
193     '&dummya=53088&datum_kz='.$anreise.'&dauer_kz='.$
194     $aufenthalt.'&pers_kz='.$personen.
195     '&kind_kz='.$kinder.'&dummys='.$sprache.'&anrede='.$anrede.'&vorname='.$vorname.
196     '&nachname='.$name.'&strasse='.$strasse.'&plz='.$plz.'&ort='.$ort.'&land='.$land.
197     '&email='.$mail.'&tel_office='.$stell.'&tel_private='.$stel2.'&tel_mobile='.$stel3.
198     '&mitteilung='.$bemerkung.'&payment='.$zahlart.'&insurance='.$rrkv.
199     '&pet='.$pet.'&travel_conditions='.$agb.$urlzusatz;
200     echo $url;

```

```

201 }
202
203 $xml_booking = @simplexml_load_file(rawurlencode($url));
204
205 if (is_object($xml_booking)) {
206 $fehler = 0;
207 /* Variante 1, Wenn bereits Eingaben oder Kartendaten inkorrekt sind */
208 foreach($xml_booking->fill_xml_order_data as $booking) {
209 $message = (string)$booking->error->errortext;
210 if ($message == 'Incomplete_Booking_Data') {
211 $fehler = 1;
212 $statusmeldung = 'Buchung_kann_nicht_bearbeitet_werden, da Angaben
213 .....fehlen.<br>Bitte_berichtigen_sie_und_versuchen_es_erneut';
214 }
215 }
216
217 /* Variante 2, wenn Daten korrekt angenommen wurden aber von Wolters ein Fehler
218 ausgeht */
219 echo '<br><br>';
220 foreach($xml_booking->bookingstate as $bookingstate) {
221 $bookstatus = (string)$bookingstate;
222 }
223 foreach($xml_booking->buchungsstatus as $booking) {
224 $status = (string)$booking;
225 if ($status != 'OK' OR $bookstatus != 'OK') {
226 foreach($xml_booking->xml_errormsg as $fehlermeldung) {
227 $fehlermeldung = (string)$fehlermeldung;
228 if ($fehlermeldung == 'AKONT.NO.FREE.KONTINGENT') {
229 $fehler = 1;
230 $statusmeldung = 'Achtung, Fehler<br>
231 .....Das_Haus_kann_im_angegebenen_Buchungszeitraum_nicht_mehr_gebucht_werden,
232 .....das_es_von_einer_anderen_Person_bereits_gebucht_wurde!';
233 }
234 }
235 }
236
237 if ($status == 'OK' AND $bookstatus == 'OK') {
238 foreach($xml_booking->buchungsnr as $buchungsnummer) {
239 $buchungsnr = (string)$buchungsnummer;
240 }
241 foreach($xml_booking->totalprice as $totalpreis) {
242 $totalpreis = (string)$totalpreis;
243 }
244 $fehler = 0;
245 $statusmeldung = 'Ihre_Buchungsanfrage_wurde_entgegen_genommen_und_wird
246 .....bearbeitet.<br>Die_Buchungsnummer_lautet: '. $buchungsnr. ' Weitere_Daten
247 .....erhalten_sie_von_Wolters_per_E-Mail<br>Viel_Spaß_im_Urlaub_&#252;nicht_ihnen
248 .....das_Team_von_Ferienhaus.de';
249 $statusmeldung = $statusmeldung. '<br>Der_Gesamtpreis_beträgt '. $totalpreis. '
250 .....Euro';
251 }
252 }
253 }
254 else {
255 $fehler = 1;
256 $statusmeldung = 'Entweder_haben_sie_schon_versucht_das_Objekt_zu_buchen,<br>oder
257 .....es_konnte_keine_Verbindung_zum_Wolters_Buchungsserver_hergestellt_werden.<br>
258 .....Bitte_versuchen_sie_es_sp&#228;ter_noch_einmal.';
259 }
260
261 if ($fehler == 1) {
262 $query = "INSERT INTO $tabelle
263 (id ,name,vorname, str ,plz ,ort ,tel ,email ,
264 anbieter ,objekt ,kategorie ,datum ,zeit ,buchungserfolg) VALUES ( ' ', '$name',
265 '$vorname', ' '$strasse', '$plz', ' '$ort', ' '$stell', ' '$mail', ' '$anbieter', '$referenz',
266 '$kategorie', '$datum_aktuell', ' '$zeit_aktuell', 'N')";

```

```

267 $result = mysql_query($query);
268 if ($result) {
269     echo 'Datensatz_eingefügt';
270 }
271 else {echo mysql_error();}
272 }
273 else {
274     $query = "INSERT INTO $tabelle_(id,name,vorname,str_plz_ort_tel_email_,
275 _anbieter,objekt,kategorie,datum,zeit,buchungserfolg)_VALUES_('','$name',
276 '$vorname','_$strasse','$plz','$sort','$stell','$mail','$sanbieter','$referenz',
277 '$kategorie','$datum_aktuell','$szeit_aktuell','Y')";
278 $result = mysql_query($query);
279 if ($result) {
280     echo 'Datensatz_eingefügt';
281 }
282 else {echo mysql_error();}
283 }
284
285 ?>
286 <html>
287   <html xmlns="http://www.w3.org/1999/xhtml">
288   <head>
289       <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
290       <title>Abfragetests</title>
291       <link href="../css/ferienhaus_ibc.css" rel="stylesheet" type="text/css">
292   </head>
293   <body>
294       <!-- ***Hauptcontainer Start*** -->
295       <div id="main">
296         <div style="width:100%; border:1px solid;">
297           <span>Adressdaten</span>&nbsp;&nbsp;&nbsp;<span>Zahlungsangaben</span>&nbsp;&nbsp;&nbsp;<span>Angaben pruefen</span>&nbsp;&nbsp;&nbsp;<span class="red">Status</span></div>
298           <?php
299             require_once '../header.php';
300           ?>
301           <div class="headline">Buchungsstatus</div>
302           <?php echo $statusmeldung; print_r($xml_booking);?>
303
304           <br>
305           <a href="#" .. / index . php">Zur&#252;ck zur Suchmaske</a><a href="#buchen1.php">
306             Zur&#252;ck zum Buchungsformular</a>
307         </div>
308       <!-- ***Hauptcontainer Ende*** -->
309     </body>
310   </html>

```

G.16. Überprüfungsroutine Buchungsstrecke

Listing G.14: Quelltext Überprüfungsscript Buchungsstrecke

```

1 <?php
2
3 /**
4  * Checkbereich
5  * Überprüfung und Verarbeitung von Formulardaten
6  * Überprüft die personenbezogenen Daten welche
7  * während der Buchung erfasst werden und ist auch
8  * für den Check der Zahlungsdaten zuständig
9  */
10
11 $anbieter = $_SESSION[ 'objekt' ][4];
12 $zeichen_illegal = array("<", ">");
13
14

```

```
15 if ($_POST['checkup'] == '1') {
16     $pruefbereich = 1;
17 }
18 elseif ($_POST['checkup'] == '2') {
19     $pruefbereich = 2;
20 }
21 elseif ($_POST['checkup'] == '3') {
22     $pruefbereich = 3;
23 }
24 elseif ($_POST['checkup'] == '4') {
25     $pruefbereich = 4;
26 }
27
28 /* Personen und Zusatzleistungen */
29 if ($pruefbereich == 1) {
30
31     $_SESSION['meldung'] = NULL;
32
33     /* Anrede */
34     if ($_POST['bu_anrede'] == '1') {
35         $_SESSION['person'][0] = 1;
36     }
37     elseif ($_POST['bu_anrede'] == '2') {
38         $_SESSION['person'][0] = 2;
39     }
40     else {
41         $fehler[0] = 'Bitte_eine_Anrede_ausw&#228;hlen';
42         $_SESSION['person'][0] = 1;
43     }
44
45     /* Name */
46     if (strpos($_POST['bu_name'], '<') OR strpos($_POST['bu_name'], '>')) {
47         $fehler[1] = 'Sie_benutzen_verbotene_Zeichen!';
48         $_SESSION['person'][1] = NULL;
49     }
50     elseif (strlen($_POST['bu_name']) > 50) {
51         $buname = substr($_POST['bu_name'], 0, 50);
52         $_SESSION['person'][1] = $buname;
53     }
54     elseif (strlen($_POST['bu_name']) < 2) {
55         $fehler[1] = 'Sie_haben_keinen_Namen_angegeben';
56         $_SESSION['person'][1] = NULL;
57     }
58     else {
59         $buname = str_replace($zeichen_illegal, "_", $_POST['bu_name']);
60         $_SESSION['person'][1] = $buname;
61     }
62
63     /* Vorname */
64     if (strpos($_POST['bu_vorname'], '<') OR strpos($_POST['bu_vorname'], '>')) {
65         $fehler[2] = 'Sie_benutzen_verbotene_Zeichen!';
66         $_SESSION['person'][2] = NULL;
67     }
68     elseif (strlen($_POST['bu_vorname']) > 50) {
69         $buvorname = substr($_POST['bu_vorname'], 0, 50);
70         $_SESSION['person'][2] = $buvorname;
71     }
72     elseif (strlen($_POST['bu_vorname']) < 2) {
73         $fehler[2] = 'Sie_haben_keinen_Vornamen_angegeben';
74         $_SESSION['person'][2] = NULL;
75     }
76     else {
77         $buvorname = str_replace($zeichen_illegal, "_", $_POST['bu_vorname']);
78         $_SESSION['person'][2] = $buvorname;
79     }
80 }
```

```

81  /* Straße */
82  if (stristr($_POST['bu_strasse'], '<') OR stristr($_POST['bu_strasse'], '>')) {
83      $fehler[3] = 'Sie_benutzen_verbotene_Zeichen!';
84      $_SESSION['person'][3] = NULL;
85  }
86  elseif (strlen($_POST['bu_strasse']) > 50) {
87      $bustrasse = substr($_POST['bu_strasse'], 0, 50);
88      $_SESSION['person'][3] = $bustrasse;
89  }
90  elseif (strlen($_POST['bu_strasse']) < 5) {
91      $fehler[3] = 'Sie_haben_keine_Strasse_angegeben';
92      $_SESSION['person'][3] = NULL;
93  }
94  else {
95      $bustrasse = str_replace($zeichen_illegal, "", $_POST['bu_strasse']);
96      $_SESSION['person'][3] = $bustrasse;
97  }
98
99  /* PLZ */
100 $buplz = $_POST['bu_plz'];
101 if (stristr($_POST['bu_plz'], '<') OR stristr($_POST['bu_plz'], '>')) {
102     $fehler[4] = 'Sie_benutzen_verbotene_Zeichen!';
103     $_SESSION['person'][4] = NULL;
104 }
105 elseif (strlen($buplz) > 10) {
106     $fehler[4] = 'Die_Postleitzahl_ist_l#228;nger_als_5_Stellen!';
107     $_SESSION['person'][4] = NULL;
108 }
109 elseif (strlen($buplz) < 4) {
110     $fehler[4] = 'Sie_haben_keine_oder_eine_ung#252;ltige_,_bzw_zu_kurze
111 Postleitzahl_angegeben';
112     $_SESSION['person'][4] = NULL;
113 }
114 else {
115     $_SESSION['person'][4] = $buplz;
116 }
117
118 /* Ort */
119 if (stristr($_POST['bu_ort'], '<') OR stristr($_POST['bu_ort'], '>')) {
120     $fehler[5] = 'Sie_benutzen_verbotene_Zeichen!';
121     $_SESSION['person'][5] = NULL;
122 }
123 elseif (strlen($_POST['bu_ort']) > 50) {
124     $buort = substr($_POST['bu_ort'], 0, 50);
125     $_SESSION['person'][5] = $buort;
126 }
127 elseif (strlen($_POST['bu_ort']) < 3) {
128     $fehler[5] = 'Sie_haben_keinen_Ort_angegeben';
129     $_SESSION['person'][5] = NULL;
130 }
131 else {
132     $buort = str_replace($zeichen_illegal, "", $_POST['bu_ort']);
133     $_SESSION['person'][5] = $buort;
134 }
135
136 /* Land */
137 if ($_POST['bu_land'] == 'kA' OR $_POST['bu_land'] == '') {
138     $fehler[6] = 'Bitte_geben_sie_ein_Land_an!';
139 }
140 elseif (strlen($_POST['bu_land']) > 3) {
141     $fehler[6] = 'Das_Landesk#252;rzel_ist_l#228;nger_als_3_Zeichen_,_bitte_mit
142 dem_Admin_in_Kontakt_treten';
143     $_SESSION['person'][6] = NULL;
144 }
145 else {
146     $buland = str_replace($zeichen_illegal, "", $_POST['bu_land']);

```

```

147     $_SESSION['person'][6] = $buland;
148 }
149
150 /* Telefon */
151 $butel = $_POST['bu_telefon'];
152 if (stristr($butel, '<') OR stristr($butel, '>')) {
153     $fehler[7] = 'Sie benutzen verbotene Zeichen!';
154     $_SESSION['person'][7] = NULL;
155 }
156 elseif (strlen($butel) > 30) {
157     $butel = substr($butel,0,30);
158     $_SESSION['person'][7] = $butel;
159 }
160 elseif (strlen($butel) < 6) {
161     $fehler[7] = 'Sie haben keine Telefonnummer angegeben';
162     $_SESSION['person'][7] = NULL;
163 }
164 elseif (!(strlen($butel) <= strlen((int)$butel)+1)) {
165     $fehler[7] = 'Sie haben keine oder eine ung#252;ltige , bzw zu kurze
166     Telefonnummer angegeben';
167     $_SESSION['person'][7] = NULL;
168 }
169 else {
170     $_SESSION['person'][7] = $butel;
171 }
172
173 /* Fax, Telefon 2 */
174 $butel2 = $_POST['bu_telefon2'];
175 if (stristr($butel2, '<') OR stristr($butel2, '>')) {
176     $_SESSION['person'][8] = NULL;
177 }
178 elseif (strlen($butel2) > 30) {
179     $butel2 = substr($butel2,0,30);
180     $_SESSION['person'][8] = $butel2;
181 }
182 elseif (strlen($butel2) < 6) {
183     $_SESSION['person'][8] = NULL;
184 }
185 elseif (!(strlen($butel2) <= strlen((int)$butel2)+1)) {
186     $_SESSION['person'][8] = NULL;
187 }
188 else {
189     $_SESSION['person'][8] = $butel2;
190 }
191
192 /* Handy, Telefon3 */
193 $butel3 = $_POST['bu_telefon3'];
194 if (stristr($butel3, '<') OR stristr($butel3, '>')) {
195     $_SESSION['person'][9] = NULL;
196 }
197 elseif (strlen($butel3) > 30) {
198     $butel3 = substr($butel3,0,30);
199     $_SESSION['person'][9] = $butel3;
200 }
201 elseif (strlen($butel3) < 6) {
202     $_SESSION['person'][9] = NULL;
203 }
204 elseif (!(strlen($butel3) <= strlen((int)$butel3)+1)) {
205     $_SESSION['person'][9] = NULL;
206 }
207 else {
208     $_SESSION['person'][9] = $butel3;
209 }
210
211 /* E-Mail */
212 $bumail = substr($_POST['bu_mail'],0,200);

```

```

213  if (stristr($bumail, '<') OR stristr($bumail, '>')) {
214      $fehler[8] = 'Sie_benutzen_verbotene_Zeichen!';
215      $_SESSION['person'][10] = NULL;
216  }
217  elseif (strpos($bumail, '@') AND strpos($bumail, '.')) {
218      $_SESSION['person'][10] = $bumail;
219  }
220  else {
221      $fehler[8] = 'Sie_haben_keine_E-Mail_Adresse_angegeben';
222      $_SESSION['person'][10] = NULL;
223  }
224
225  /* Personenanzahl */
226  $buperso = (int)$POST['bu_personen'];
227  if (strlen($POST['bu_personen']) < 3 AND $POST['bu_personen'] != '' AND
228  $POST['bu_personen'] != '0') {
229      $_SESSION['person'][11] = $buperso;
230  }
231  else {
232      $fehler[9] = 'Sie_haben_keine_g#252;ltige_Anzahl_an_Personen_angegeben!';
233      $_SESSION['person'][11] = NULL;
234  }
235
236  /* Kinderanzahl */
237  $bukind = (int)$POST['bu_kinder'];
238  if (strlen($POST['bu_kinder']) < 3 AND $POST['bu_kinder'] != '' AND
239  $POST['bu_kinder'] != '0') {
240      $_SESSION['person'][12] = $bukind;
241  }
242  else {
243      $_SESSION['person'][12] = 0;
244  }
245
246  /* Zahlungsart */
247  $paymode = $POST['zahlungsart'];
248  if ($paymode == '1') {
249      $_SESSION['payment'][0] = 1;
250  }
251  elseif ($paymode == '2') {
252      $_SESSION['payment'][0] = 2;
253  }
254  elseif ($paymode == '3') {
255      $_SESSION['payment'][0] = 3;
256  }
257
258  /******
259  /* Interhome spezifische Angaben */
260  /******
261  if ($anbieter == '1') {
262
263      /* Reise Rücktritts Kostenschutz */
264      $rrkv = $POST['rrkv'];
265      $rrkv = explode("-", $rrkv);
266      $_SESSION['zusatz'][0] = $rrkv;
267      /* Freiwillige Zusatzleistungen flushen */
268      for ($i = 0; $i < 30; $i++) {
269          $_SESSION['zusatz'][$i+5] = NULL;
270      }
271      /* Freiwillige Zusatzleistungen */
272      /* —> Noch codierung der Leistungsarten einf#252;hren */
273      for ($i = 0; $i < 30; $i++) {
274          $leistung = $POST[$i];
275          if (isset($leistung)) {
276              $zusatzleistung = explode("-", $leistung);
277              $_SESSION['zusatz'][$i+5] = array($zusatzleistung[0], $zusatzleistung[1],
278              $zusatzleistung[2], $zusatzleistung[3],1, $zusatzleistung[4]);

```



```

279     }
280 }
281
282 $agb = $_POST['agb'];
283 if ($agb !== 'Y') {
284     $fehler[11] = 'Bitte_best&#228;tigen_sie_die_AGB\'s';
285     $_SESSION['person'][15] = NULL;
286 }
287 else {
288     $_SESSION['person'][15] = 'Y';
289 }
290 }
291
292 /*****
293 /* Wolters spezifische Angaben */
294 /*****
295 if ($anbieter == '2') {
296
297     /* Haustier als Merkmal */
298     $haustiermm = $_POST['H2'];
299     if ($haustiermm >= 1 AND $haustiermm <= 6) {
300         $_SESSION['zusatz'][4] = array('H2', 'Haustier', 0, 'EUR', $haustiermm, NULL);
301     }
302     else {
303         $_SESSION['zusatz'][4] = NULL;
304     }
305
306     for ($i = 0; $i < 30; $i++) {
307         $leistung = $_POST[$i];
308         $leistung_array = explode("-", $leistung);
309         if ($leistung_array[0] >= 1 AND $leistung_array[0] <= 6) {
310             $leistung_beschr = utf8_decode($leistung_array[1]);
311             $query = "SELECT_IDZusatz_FROM_ref_zusatzleistungen_WHERE_Beschreibung_LIKE
312 ..... '$leistung_beschr'_LIMIT_1";
313             $result = mysql_query($query);
314             if (mysql_num_rows($result) < 1) {
315                 $zusatzcode = 0;
316             }
317             else {
318
319                 while($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
320                     $zusatzcode = $line['IDZusatz'];
321                 }
322             }
323
324             $_SESSION['zusatz'][$i+5] = array($zusatzcode, $leistung_array[1],
325             $leistung_array[2], $leistung_array[3], $leistung_array[0],
326             $leistung_array[4]);
327         }
328         else {
329             $_SESSION['zusatz'][$i+5] = NULL;
330         }
331     }
332
333     /* Buchungshinweise */
334     $buhinweise = $_POST['buhinweise'];
335     if ($buhinweise == 'Y') {
336         $_SESSION['person'][14] = 'Y';
337     }
338     else {
339         $fehler[12] = 'Sie_m&#252;ssen_die_Buchungshinweise_best&#228;tigen!';
340         $_SESSION['person'][14] = 'N';
341     }
342
343     /* AGBs */
344     $agb = $_POST['agb'];

```

```

345     if ($agb == 'Y') {
346         $_SESSION['person'][15] = 'Y';
347     }
348     else {
349         $fehler[13] = 'Sie müssen die Allgemeinen Geschäftsbedingungen von
350         Wolters akzeptieren';
351         $_SESSION['person'][15] = 'N';
352     }
353
354     /* Reise Rücktritts Kostenschutz */
355     if ($anbieter == '2') {
356         if ($_POST['rrkv'] == 'Y') {
357             $_SESSION['zusatz'][0] = 'Y';
358         }
359         else {
360             $_SESSION['zusatz'][0] = 'N';
361         }
362     }
363
364     /* Zusatztext */
365     if ($anbieter == '2') {
366         if (strpos($_POST['bu_zusatztext'], '<') OR strpos($_POST['bu_zusatztext'],
367         '>')) {
368             $fehler[10] = 'Sie benutzen verbotene Zeichen!';
369             $_SESSION['person'][13] = NULL;
370         }
371         elseif (strlen($_POST['bu_zusatztext']) > 500) {
372             $buzust = substr($_POST['bu_zusatztext'], 0, 500);
373             $_SESSION['person'][13] = $buzust;
374         }
375         else {
376             $buzust = str_replace($zeichen_illegal, "", $_POST['bu_zusatztext']);
377             $_SESSION['person'][13] = $buzust;
378         }
379     }
380 }
381 }
382
383 if ($fehler) {
384     $_SESSION['meldung'] = $fehler;
385     Header("Location:buchen1.php");
386 }
387
388 }
389
390 /* Zahlung "Bankeinzug" */
391 if ($pruefbereich == 2) {
392
393     $_SESSION['meldung'] = NULL;
394
395     /* Kontonummer */
396     $kto = (int)$_POST['kto'];
397     if (strlen($kto) > 5 AND strlen($kto) < 13) {
398         $_SESSION['payment'][1][0] = $kto;
399     }
400     else {
401         $fehler[0] = 'Kontonummer zu kurz oder zu lang';
402         $_SESSION['payment'][1][0] = NULL;
403     }
404
405     /* Bankleitzahl */
406     $blz = (int)$_POST['blz'];
407     if (strlen($blz) > 4 AND strlen($blz) < 9) {
408         $_SESSION['payment'][1][1] = $blz;
409     }
410     else {

```

```

411     $fehler[1] = 'Bankleitzahl_zu_kurz_oder_zu_lang';
412     $_SESSION['payment'][1][1] = NULL;
413 }
414
415 /* Bank */
416 $bank = $_POST['bank'];
417 if (strlen($bank) > 7) {
418     $_SESSION['payment'][1][2] = $bank;
419 }
420 else {
421     $fehler[2] = 'Bitte_vollst&#228;ndigen_Banknamen_angeben';
422     $_SESSION['payment'][1][2] = NULL;
423 }
424
425 if ($fehler) {
426     $_SESSION['meldung'] = $fehler;
427     Header("Location:buchen2.php");
428 }
429 }
430
431 /* Zahlung Kreditkarte */
432 if ($pruefbereich == 3) {
433
434     $_SESSION['meldung'] = NULL;
435
436     /* Kreditkartentyp */
437     $cctype = $_POST['cctype'];
438     /* *****/
439     /* Interhome spezifische Angaben */
440     /* *****/
441     if (strlen($cctype) == 4) {
442         $_SESSION['payment'][2][0] = $cctype;
443     }
444     /* *****/
445     /* Wolters spezifische Angaben */
446     /* *****/
447     if (strlen($cctype) == 2) {
448         $_SESSION['payment'][2][0] = $cctype;
449     }
450
451     /* Kreditkartennummer */
452     $ccnum = (int)$_POST['ccnum'];
453     if (strlen($ccnum) < 16 OR strlen($ccnum) > 20) {
454         $fehler[0] = 'Nummer_ung&#252;ltig';
455         $_SESSION['payment'][2][1] = NULL;
456     }
457     else {
458         $_SESSION['payment'][2][1] = $ccnum;
459     }
460
461     /* Verifizierungsnummer */
462     $ccver = (int)$_POST['ccver'];
463     if (strlen($ccver) == 3) {
464         $_SESSION['payment'][2][2] = $ccver;
465     }
466     else {
467         $fehler[1] = 'Nummer_ung&#252;ltig';
468         $_SESSION['payment'][2][2] = NULL;
469     }
470
471     /* Ablaufdatum Jahr */
472     $ccexpirey = (int)$_POST['ccexpirey'];
473     if ($ccexpirey >= date("Y") AND strlen($ccexpirey) == 4) {
474         $_SESSION['payment'][2][3] = $ccexpirey;
475     }
476     else {

```

```

477     $fehler[2] = 'Karte_ist_abgelaufen';
478     $_SESSION['payment'][2][3] = NULL;
479 }
480
481 /* Ablaufdatum Monat */
482 $ccexpirem = (int)$_POST['ccexpirem'];
483 if ($ccexpirey == date("Y") AND $ccexpirem >= date("n") AND strlen($ccexpirem) <
484 3) {
485     $_SESSION['payment'][2][4] = $ccexpirem;
486 }
487 elseif ($ccexpirey > date("Y") and strlen($ccexpirem) < 3) {
488     $_SESSION['payment'][2][4] = $ccexpirem;
489 }
490 else {
491     $fehler[2] = 'Karte_ist_abgelaufen';
492 }
493
494 /* Name */
495 $cclname = $_POST['cclname'];
496 if (strpos($cclname, '<') OR strpos($cclname, '>')) {
497     $fehler[3] = 'Sie_benutzen_verbotene_Zeichen!';
498     $_SESSION['payment'][2][6] = NULL;
499 }
500 elseif (strlen($cclname) > 50) {
501     $cclname = substr($cclname,0,50);
502     $_SESSION['payment'][2][6] = $cclname;
503 }
504 elseif (strlen($cclname) < 2) {
505     $fehler[3] = 'Sie_haben_keinen_Namen_angegeben';
506     $_SESSION['payment'][2][6] = NULL;
507 }
508 else{
509     $cclname = str_replace($zeichen_illegal, "", $cclname);
510     $_SESSION['payment'][2][6] = $cclname;
511 }
512
513 /* Vorname */
514 $ccfname = $_POST['ccfname'];
515 if (strpos($ccfname, '<') OR strpos($ccfname, '>')) {
516     $fehler[4] = 'Sie_benutzen_verbotene_Zeichen!';
517     $_SESSION['payment'][2][5] = NULL;
518 }
519 elseif (strlen($ccfname) > 50) {
520     $ccfname = substr($ccfname,0,50);
521     $_SESSION['payment'][2][5] = $ccfname;
522 }
523 elseif (strlen($ccfname) < 2) {
524     $fehler[4] = 'Sie_haben_keinen_Vornamen_angegeben';
525     $_SESSION['payment'][2][5] = NULL;
526 }
527 else {
528     $ccfname = str_replace($zeichen_illegal, "", $ccfname);
529     $_SESSION['payment'][2][5] = $ccfname;
530 }
531
532 if ($fehler) {
533     $_SESSION['meldung'] = $fehler;
534     Header("Location:buchen2.php");
535 }
536 }
537 }
538
539 /* Bezahlung auf Rechnung */
540 if ($pruefbereich == 4) {
541
542     $_SESSION['meldung'] = NULL;

```

```
543
544 $rechnung = $_POST['rechnung'];
545 if ($rechnung == 'Y') {
546     $_SESSION['payment'][3][0] = $rechnung;
547 }
548 else {
549     $fehler[0] = 'Bitte bestätigen sie die Zahlung auf Rechnung.';
550     $_SESSION['payment'][3][0] = NULL;
551 }
552
553 if ($fehler) {
554     $_SESSION['meldung'] = $fehler;
555     Header("Location:buchen2.php");
556 }
557 }
558 ?>
```

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Alle Teile, die wörtlich oder sinngemäß einer Veröffentlichung entstammen, sind als solche kenntlich gemacht.

Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.

Berlin, den 29.07.2009